Universität Rostock

Fakultät für Informatik
und Elektrotechnik

A General Framework for Digital Game-Based Training Systems

Dissertation
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik
der Universität Rostock

vorgelegt von
Angela Brennecke, geb. am 23. Januar 1978 in Braunschweig
aus Rostock

Rostock, 31. März 2009

urn:nbn:de:gbv:28-diss2009-0152-0

Approved by

Prof. Dr. Heidrun Schumann, University of Rostock, Germany (Adviser)
Prof. Dr. Pavel Slavik, Czech Technical University Prague, Czech Republic
Prof. Dr. Stefan Schlechtweg, Anhalt University of Applied Sciences (FH), Germany

Day of Defense

July 09, 2009

Keywords

Computer Graphics, Virtual Reality, Digital Game-Based Learning, Digital Game-Based Training Systems, Serious Games

urn:nbn:de:gbv:28-diss2009-0152-0

Abstract

Serious games have recently become the most prominent example of digital game-based learning and digital game-based training systems. However, the integration of serious games into education has failed so far. This is mainly due to the fact that serious games pursue traditional computer game design patterns and address only learning situations. Yet apart from learning situations, education is comprised of teaching and communication situations that let teacher and student interact with each other. This dissertation presents a general framework for developing digital game-based training systems that addresses all aspects of education to overcome this problem. The framework appeals to both teacher and student, as game players. It is very well suited to adapt training content to the lessons as well as to integrate existing learning material into the training system. A visual review log summarizes the training sessions and is intended to be used for classroom discussions. The general framework is accompanied by a practical workflow schedule, which breaks down the complexity of developing digital game-based training systems into four well-structured stages. Based on the workflow schedule, a novel digital game-based training system for crime scene investigation (CSI) training called *OpenCrimeScene* has been developed and implemented based on the general framework. A system evaluation confirms the usefulness of the concepts developed here.

Zusammenfassung

Serious Games ("ernste Spiele") sind in letzter Zeit zu dem bekanntesten Beispiel des computerspielbasierten Lernens und der computerspielbasierten Trainingssysteme geworden. Jedoch ist die Integration von Serious Games in die Ausbildung bisher fehlgeschlagen. Dies liegt hauptsächlich darin begründet, das Serious Games die Designmuster herkömmlicher Computerspiele verfolgen und nur Lernsituationen ansprechen. Allerdings besteht die Ausbildung neben Lernsituationen aus Lehr- und Kommunikationssituationen, in denen Lehrer und Schüler miteinander interagieren. Diese Dissertation stellt ein allgemeines Rahmenwerk für die Entwicklung von computerspielbasierten Trainingssystemen vor, welches sich an alle Aspekte der Ausbildung richtet, um dieses Problem zu überwinden. Das Rahmenwerk spricht beide, Lehrer und Schüler, als Spieler an. Es ist besonders gut geeignet, den Trainingsinhalt an den Unterricht anzupassen sowie vorhandene Lehrmaterialien in das Trainingssystem einzubinden. Ein grafisches Bewertungsprotokoll fasst die Trainingseinheiten zusammen und soll für Diskussionen im Unterricht genutzt werden. Das allgemeine Rahmenwerk wird von einem Arbeitsablaufsplan begleitet, der die Komplexität der Entwicklung von computerspielbasierten Trainingssystemen in vier gut strukturierte Abschnitte untergliedert. Der Arbeitsablaufsplan dient als Grundlage für die Entwicklung und Implementierung eines neuartigen computerspielbasierten Trainingssystems für Tatortsicherungstraining namens *OpenCrimeScene*, das auf dem allgemeinen Rahmenwerk basiert. Eine Evaluierung des Systems bestätigt die Nützlichkeit der hier entwickelten Konzepte.

Danksagung

Mein herzlicher Dank gilt Prof. Dr. Thomas Strothotte, der mir diese Promotion zunächst an der Otto–von–Guericke Universität Magdeburg, später an der Universität Rostock, ermöglicht hat. Insbesonders möchte ihm dafür danken, dass er trotz eigener beruflicher Neuorientierung stets für die weitere inhaltliche Betreuung wie auch finanzielle Absicherung dieser Arbeit gesorgt hat. Gleichermaßen möchte ich Prof. Dr. Stefan Schlechtweg danken, dass er die Betreuung meiner Arbeit fortgeführt und die Leitung des Projekts “Der virtuelle Tatort” übernommen hat. Somit konnte ich den Fokus meiner zunächst sehr praktisch-orientierten Arbeit auf den wissenschaftlichen Inhalt lenken.

Ganz besonderer Dank gilt Prof. Dr. Heidrun Schumann, die die Betreuung meiner Arbeit an der Universität Rostock übernommen und mir dabei geholfen hat, den zunächst sehr breit angelegten Inhalt der Arbeit in eine klare Form zu gießen. Ich danke ihr für viele aufschlussreiche Diskussionen, durch die ich nicht nur im fachlichen Bereich neue Ideen entwickeln konnte. Insbesondere möchte ich ihr auch dafür danken, dass sie mich so herzlich in ihre Arbeitsgruppe aufgenommen hat. Dieser Dank gilt gleichermaßen all meinen Kollegen, die mir jegliche Zweifel an der Entscheidung, nach Rostock zu gehen, genommen haben. Sehr herzlich danke ich in diesem Zusammenhang Tobias Germer für die gemeinsame Zeit in unserem Magdeburger Büro, die für mich zu einer bleibenden Erinnerung geworden ist. Ebenso danke ich auch den anderen Kollegen aus meiner Magdeburger Arbeitsgruppe für ihre Unterstützung. Prof. Pavel Slavik danke ich, dass er die Begutachtung meiner Arbeit ohne zu zögern übernommen hat.

Im Zusammenhang mit dem Projekt “Der virtuelle Tatort” gilt mein besonderer Dank Kriminaldirektor Ralf Heidrich und Polizeihauptkommissar Peter Eichardt, mit denen ich während des Projekts eng zusammengearbeitet habe. Ihre fachliche Unterstützung, die Bereitstellung jeglicher Art von Informationsmaterial sowie ihre großen Bemühungen, Computervisualisten mit der Tatortsicherung vertraut zu machen, haben diese Dissertation maßgeblich unterstützt.

Sehr herzlich möchte ich mich bei Jay Knox bedanken, der die gesamte Arbeit Korrektur gelesen und mir bei vielen fachlichen Fragen zur Pädagogik und Ausbildung zur Verfügung stand. Ebenfalls danken möchte ich den vielen Studenten, die zur Entwicklung von *OpenCrimeScene* beigetragen haben. Hierzu gehören Tom Brosch, Jens Grubert, Ramona Grzeschik, Thomas Habener, Alexander Hewicker, Sebastian Hundt, Robert Maertins, Marie-Luise Müller, Christian Panzer, Kerstin Reinemann, Nicole Reinke, Maik Schulze, Thomas Seidl, René Sickel und Michael Spengler.

Meiner Familie und meinen Freunden möchte ich ganz herzlich für ihre Unterstützung während meiner Promotion — insbesondere in den letzten Wochen—danken. Ich bin sehr froh und glücklich, dass es Euch alle gibt!

Last but not least, I would like to thank Kamran Ali for joining me in Rostock and for having such a great time — struggling with the PhD, relaxing at the beach, and dreaming about the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Current trends in digital game-based learning research show an increase in serious game development and use. Serious games present learning objectives in game-like environments and tightly connect learning and training to computer game playing [Michael and Chen, 2006]. Two factors are crucial for this: first, computer games interactively involve the player in the game world and game story. Second, computer games keep the player's motivation to continue playing at a high level throughout the game. This is significantly desirable for education and makes serious games attractive for learning [Gee, 2003; Prensky, 2001].

Yet serious game have failed to become a successful tool in education [Susi et al., 2007; van Eck, 2006]. Most applications are built from scratch and pursue traditional game design patterns. As a consequence, serious games seldom relate to educational settings. They do not address teachers as users, nor do they explicitly link game content to lessons or existing learning material. On the one hand, researchers stress that not every game suits every educational situation; on the other hand, researchers stress that games have to be *integrated into* rather than simply *used during* the lessons [Capps, 2002; Harteveld and Bidarra, 2007; van Eck, 2006]. However, the question about how to achieve this has not been resolved so far and novel approaches to serious game development are called for [FAS and ESA, 2006; Kelly, 2005; van Eck, 2006].

1.1 Clarification of the Problem & Goals

The increasing success of computer games over the last couple of decades has been ascribed to several game features. Among them are the mixture of challenge & reward, gripping game stories that immerse the players, competition, and compelling graphical user interface design. In a successful game, all of these aspects are combined to realize the potential to motivate the player to continue playing. This potential has made bad headlines only recently when the Criminological Research Institute of Lower Saxony published the results of a large study on the computer game playing behavior of teenage pupils [Rehbein et al., 2009]. The study revealed that 20.1 % of the 44 610 participants of the study play computer games extensively and that 3.3 % of the participants are even considered to be addicted to computer game playing. This is of course the downside of successful computer game development. Yet at the same time it shows how strong the effect of these software applications can become — even though to a negative extent.

The research field of digital game-based learning tries to use the potential of computer games as a potential to motivate students to continue learning. So-called *serious games* are currently the predominant application type of digital game-based training systems that are used to support digital game-based learning strategies — yet with mixed success. Serious games have a strong computer game background and, thus, pursue computer game design patterns. The applications commonly address students as game players and provide a game story that is situated in the specific educational field. This is suitable to enhance learning processes. However, the goal of turning serious games into a regular educational supplement and integrating the software applications into classrooms has failed so far. The main reason for this is that serious games are restricted to enhancing learning situations, but they do not adequately address the other elements of education.

Education can typically be described by three relationships between teacher, student, and learning content that represent teaching, learning, and communication situations. As was outlined above, serious games are used to enhance learning situations, but neither do they address teaching situations nor do they support communication between teacher and student. This hinders their integration into education. Further difficulties can be summarized as follows: teachers are generally not familiar with computer games, which makes it difficult for them to instruct students how to use the applications. Moreover, serious games usually provide fixed content, which does not necessarily relate to the lessons and can hardly be adapted to them. As a consequence, students discredit the educational use of serious games, especially when the applications favor fiction over facts. Finally, serious games lack adequate assessment strategies, which could be used in the classrooms to discuss the game playing experiences [Chen and Michael, 2005; Kelly, 2005]. Since all of these issues are somehow interwoven, use of serious games in education has been retarded.

The main objective of this dissertation is to provide a solution to overcome these issues. The following list of goals will be targeted here in order to achieve this:

Developing a general framework for digital game-based training systems. Central to this dissertation is the development of a general framework for digital game-based training systems that addresses all aspects of education. Digital game-based training systems have to supplement and not replace traditional educational situations. The following aspects have to be kept in mind for this:

- Teachers have to be explicitly invited to use such applications and have to be acquainted with digital game-based training systems.
- Students have to be assured that the presented content relates to real learning material.
- Training content has to be adapted to the lessons.
- Assessment material has to be provided by the systems that can be used to support teacher-student-communication.

All in all, digital game-based training systems have to supplement and improve but not replace traditional educational situations.

Establishing a workflow schedule. Development and implementation of serious games and digital game-based training systems is highly complex. Even though instructional design patterns exist that target educational software, a novel framework approach will require a precise workflow schedule. The workflow schedule therefore has to support the developers of digital game-based training systems.

Implementing a proof of concept. A prototypical digital game-based training systems has to be developed based on the general framework to verify the approach. Therefore, close collaboration with an educational field will be required. Furthermore, a system evaluation has to conclude the development.

By tackling these issues, the dissertation contributes to the integration of digital game-based learning into education. Yet in addition, this dissertation also tries to acquaint adults with this kind of technology in order to better understand the methodologies of computer games and consequently to better understand young game players, which might help to prevent abuse of computer games.

1.2 Contributions

The following list of results presents the central contributions made by this dissertation to the research field of digital game-based learning in order to support the integration of digital game-based training systems into education:

A general framework for digital game-based training systems was developed that addresses all aspects of education. The general framework contains an authoring, training, and reviewing component that technically represent and support teaching, learning, and communication & interaction situations. The authoring and training components allow teachers to specify training content and students to engage with training content. The reviewing component assesses training sessions and generates a visual review log that can be used as assessment material for classroom discussion. A workflow schedule accompanies the general framework to facilitate the development and implementation of digital game-based training systems. A publication about the general framework was accepted for *IADIS Game and Entertainment 2009* and was considered to be relevant by all reviewers. Details can be found in [Brennecke and Schumann, 2009].

New concepts were developed for authoring and experiencing digital game-based training systems. The interaction paradigms of *in-game authoring* and *in-game training* have been developed. In-game authoring turns the authoring process into a game playing experience and supports non-expert users to specify training content. In-game training targets the integration of existing learning material into the training system and promotes authentic data presentation. Both aspects are regarded to assure students that the training system is relevant for education. In that regard, the traditional game story concept was split into a back-story and a main-story. Both address teacher and student as game players, respectively. While playing the back-story, the teacher can specify training content, which serves as a basic input for the main-story. This way, teachers can participate in the fun-part of game playing and pursue teaching situations at the same time. On the other hand, students are reassured that playing the main-story is an adequate training strategy. They might be additionally motivated to use digital game-based training systems if they have to play against the teacher. The development of a virtual SLR camera training tool demonstrated the concept of in-game training and was published as a journal paper in [Brennecke et al., 2008a].

The novel concept of a visual review log was introduced to comprehensibly summarize training sessions. The visual review log is a summary of training sessions presented in the form of a short story or overview illustration. The main target of the visual review log is to support teacher-student-communication in the classroom. It is intended to serve as a basis for discussion as well as for the teacher's evaluation or self-evaluation. Therefore, user interaction has to be logged, analyzed, and visualized. One solution to visualize user interaction in an interactive 3D environment would be to stylize outlines of the player character and scene objects. The concept of G-strokes simplifies line stylization rendering and could, thus, well be applied for stylizing character outlines. The fact that G-strokes can also be turned into vector graphics makes them additionally suitable for the visual review log. The concept of the visual review log was published in [Brennecke et al., 2007]. The concept of G-strokes was published as a journal paper in [Isenberg and Brennecke, 2006]. Their application to vector graphics was published in [Isenberg et al., 2005].

A digital game-based training system for crime scene investigation training has been developed and implemented. Called *OpenCrimeScene*, the system verifies the applicability of the general framework to an actual implementation and substantiates the usefulness of the workflow schedule. *OpenCrimeScene* reproduces typical crime scene investigation training procedures and was developed in close collaboration with the Police College of Saxony–Anhalt, Germany. The prototype has been described and published in [Brennecke et al., 2008b].

A first evaluation of the novel concepts developed here as well as the *OpenCrimeScene* prototype led to very positive results. A large number of diploma theses and student assignments were supervised during the scope of this dissertation, which contributed to the implementation of *OpenCrimeScene*. A list of the written assignments can be found in Appendix A.

1.3 Outline of the Dissertation

The dissertation is structured as follows:

Chapter 2 introduces the reader to the historical development of learning environments and computer games that has culminated in the software type of serious games to allow for a deeper understanding of the goals of this dissertation. Relevant terms and definitions are summarized next. Subsequently, the current state of the art of training environments and serious games is presented together with recent studies on using serious games in education. The final point of this chapter is the presentation of crime scene investigation training at the Police College of Saxony–Anhalt, which will serve as an application scenario here.

Chapter 3 presents the development of the general framework for digital game-based training systems. Therefore, two schematic representations are established and related that describe educational situations and game playing settings. These are referred to as *educational universe* and *game universe*. The game universe is then extended to match the elements of the educational universe and serves as a basis for the general framework for digital game-based training systems. The framework consists of three main components for authoring, training, and reviewing. These technically represent educational situations and support teaching, learning, and communication, respectively. A game context is furthermore created that introduces the concepts of back-story and main-story to digital game-based learning.

Chapter 4 discusses implementation issues that result from the general framework. In the context of digital game-based learning, authentic data presentation is one first challenge that has to be faced when implementing a digital game-based training system. Moreover, the authoring, training, and reviewing components pose further challenges to the implementation. Content authoring has to be made available to non-experts; digital training has to be closely related to real world education; reviewing has to result in a comprehensible visual review log that can support classroom discussion. Based on these considerations, a workflow schedule is formed that serves as a basic structure for implementing digital game-based training systems.

Chapter 5 follows the workflow schedule and describes the application of the general framework to police education as well as the implementation of a digital game-based training system for CSI training called *OpenCrimeScene*. The development and implementation of the individual framework components as well as of the data presentation for *OpenCrimeScene* are discussed in detail and a final system overview is presented. The chapter closes with a description of the system evaluation and an examination of the evaluation results.

Chapter 6 summarizes the results and contributions made in this dissertation and provides a brief discourse on future work.

Chapter 2

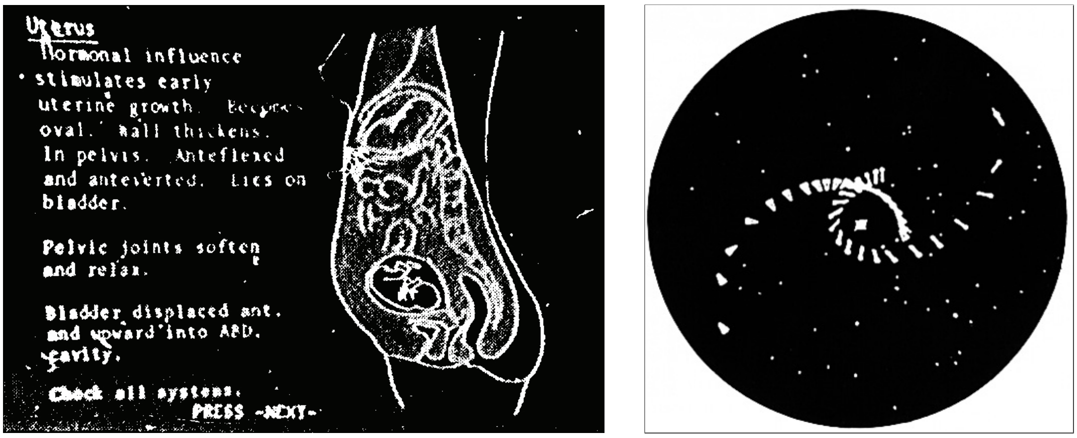
Computer Games for Learning

The reason for using computer games for learning is that computer games —in addition to their entertaining qualities—evoke learning processes [Crawford, 1984; Prensky, 2001]. As phrased by Egenfeldt-Nielsen, "A very basic premise of playing the majority of computer games is to engage with an unknown universe, slowly learning more about this universe," [Egenfeldt-Nielsen, 2005, p. 10]. Research on linking this universe with learning content is known as *digital game-based learning* with serious games being currently the software type predominantly used [van Eck, 2006].

The present chapter examines the research field of digital game-based learning and digital game-based training systems in greater detail. To start with, the history of educational software and computer games is briefly presented (cf. Sec. 2.1). Subsequently, relevant terms and definitions are reconsidered for a deeper understanding of digital game-based learning (cf. Sec. 2.2). The state of the art of digital game-based learning and digital game-based training systems is examined next. Benefits and drawbacks of current digital game-based training systems will be thoroughly discussed (cf. Sec. 2.3). Finally, the application scenario of crime scene investigation training is presented to exemplify requirements of practical training and higher education (cf. Sec. 2.4).

2.1 Historical Development

Research on digital game-based learning has been done for about ten years now. The history of training systems and computer games is far older and closely connected. The roots of present serious games reach back to the early 1960s. The first computer system for educational use was developed by Donald Bitzer at the University of Illinois in 1962 called PLATO (*Programmed Logic for Automated Teaching Operations*) [Bitzer et al.,



(a) [Bitzer et al., 1970]

(b) [Gratz, 1981]

Figure 2.1.: Visualization of the female uterus for a nursing lesson of the PLATO system (a). Screenshot taken from the original *Spacewar!* computer game (b). The spaceships are the small white polygons (called ”Wedge“ and ”Needle“), which were steered by the users to pass the surrounding stars and asteroids (white dots).

1961, 1962]. The system's training unit provided drill & practice exercises. Users could take virtual lessons sequentially and repetitively; lessons were stored on a mainframe computer and could be accessed through terminals [Bitzer et al., 1966; Wolley, 1994]. PLATO managed different user roles, for example, *student* for exercising, *instructor* for examining exercises, and *author* for creating exercises. In addition, PLATO allowed for sharing educational content and provided communication means like forums, chat, and email via a so-called *electronic blackboard* [Bitzer et al., 1966; Kearsley, 1982; Wolley, 1994]. The learning content covered nearly every educational field (see Fig. 2.1(a)).

Apart from learning, PLATO's networking capabilities were used for playing computer games. One of these games was the first so-called computer game ever: *Spacewar!* It had been developed at the Massachusetts Institute of Technology (MIT) by Steve Russel et al. in the early 1960s as demo to test a new mainframe computer [Brand, 1972; Gratz, 1981]. *Spacewar!* was a two-player game with fighting spaceships and even contained gravity effects (see Fig. 2.1(b)). Due to the lack of standards for computer hardware and software the game could not be ported to PLATO until 1969 [Wolley, 1994].

2.1.1 Using Computers for Learning

While the development of computer games soon turned into its own industry, research on computers for learning developed into a discrete academic field. By the end of the

1960s the first computer science departments focusing on computers and education were formed. The Learning Research Group at Xerox PARC in Palo Alto, California, was led by Alan Kay [Gasch, 2005]. Before Kay became head of the Learning Research Group at Xerox PARC, he had been working together with Seymour Papert, who promoted the use of computers in children's education as a creative tool. Papert pursued the idea that learning could best be supported if the learner reconstructed meaningful objects. This contrasted with existing drill & practice systems like the PLATO system. Papert called his theory *Constructionism*.[1] It was supplemented with the interpretive *Logo* programming language and the *Turtle*, a cybernetic animal that supported *Logo* graphically, as illustrated in Figure 2.2(a) [Harel and Papert, 1991; Papert, 1999]. In addition, Papert introduced the concept of *MicroWorlds*. These are virtual environments for exploring and reconstructing educational content [Papert, 1980, 1987].

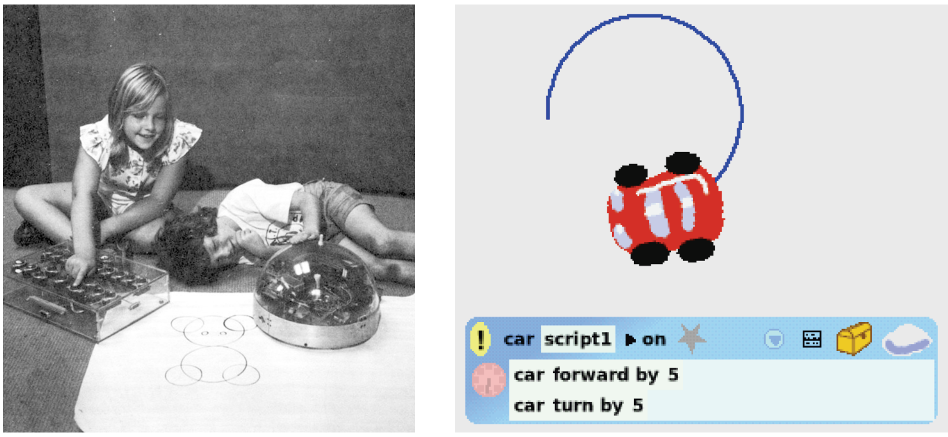
Based on Papert's work, Kay developed further concepts that supported computer use for non-professionals, in particular children. One result was the *Smalltalk* programming language, an object-oriented and dynamically typed language [Frenkel, 1994; Kay, 1993]. *Smalltalk* was implemented in an interactive graphics-based programming environment that introduced overlapping windows and opaque pop-up menus [Kay, 1993]. *Smalltalk* was later re-implemented. Called *Squeak Smalltalk*, it was used to implement two complex development frameworks for learning and collaboration: *Squeak Etoys* and *Croquet* [Ingalls et al., 1997; Kay, 2005; Smith et al., 2004]. Especially *Squeak Etoys* draws on Papert's ideas and extensively uses visual objects and multimedia elements to acquaint children with computers and programming (see Fig. 2.2(b)).

Speaking of visual objects and multimedia elements, in particular gaming industry pushed ahead with developing realistic computer graphics that could be dealt with interactively and in real-time.

2.1.2 Using Computers for Game Playing

After having done research at Xerox Parc for about 10 years Alan Kay went on to work at Atari Inc. in 1980 [Gasch, 2005]. Atari Inc. had been one of the first companies to produce computer games for the arcades and had a major success with the two player table-tennis

1 Papert's theory drew on Jean Piaget's *Constructivism* as a result of a close collaboration between Papert and Piaget during the 1960s.



(a) [Papert, 1980]

(b) [Kay, 2005]

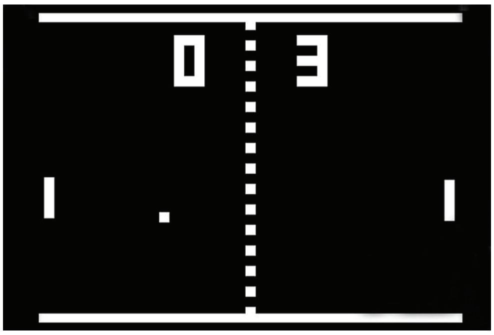
Figure 2.2.: Two children controlling the floor Turtle to generate a line drawing of a Teddy bear that consists of eight circles at different sizes (a). The Turtle's movement can be controlled via a *Logo* program. It is captured in a line drawing. In order to generate a particular line drawing children have to deal with mathematical concepts like distances and rotations. *Squeak Etoys* consists of an easy-to-use scripting language for programming (b). This can be used to animate 2D graphics objects like the car, similar to the *Logo* programming.

game *Pong*.[2] Released in 1972, Pong only consisted of simple animated graphics (see Fig. 2.3(a)). Yet with the growing success of computer games, more realistic graphics were developed. Notably rendering of computer graphics in a *first-person perspective* contributed to increase the game's *immersion depth*.[3] An early approach to this rendering perspective is illustrated in Figure 2.3. It is commonly referred to as *first-person shooter* or simply *FPS* perspective due to its predominant use in that genre of computer games.

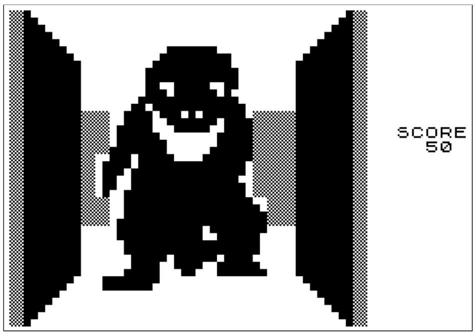
In 1978 Atari Inc. entered the home computer market to reach even more people [Bogost, 2006; Linzmayer, 1999]. According to Cohen [1984], Atari became the fastest growing company in U. S. history at the time — with sales of $ 2 billion and profits of over $ 300 million in 1982. Even the North American video game crash (1983–1985), causing losses of $ 500 million, could not impede the ongoing success [Ahl, 1984]. The success of computer games has not declined since. In 2007, U. S. computer game software sales alone reached $ 9.5 billion, with total sales of $ 18.8 billion for software, hardware, and accessories [NPD Group Inc., 2008]. An increase of 15 % was recorded for 2008 [NPD Group Inc., 2009]. Compared to that, German computer games software sales

2 Atari Inc. was founded by Nolan Bushnell and Ted Dabney in 1972. Both had developed the first commercial coin-operated arcade computer game *Computer Space*, another clone of the original *Spacewar!* game, but had sold it to the arcade game manufacturer Nutting Associates in 1971. One year later they decided to start their own business [Cohen, 1984; Fulton, 2007].

3 Immersion depth is the extent to which players feel emotionally involved in the game.



(a) Pong



(b) 3D Monster Maze

Figure 2.3.: Screenshot of Atari's table tennis game *Pong* (a). The bars on the left- and right-hand side represent the rackets that could be moved up and down to hit the ball (white quad). *3D Monster Maze* by J. K. Greye Software was one of the first computer games for home computers that applied so-called 2 ½D raster graphics (b).

reached € 1.4 billion in 2007 and € 1.6 billion in 2008 [BIU, 2009]. These figures not only indicate the popularity of computer games, but also imply that game development will be continued on a large budget.

As will be described next, it was not interactive graphics alone that made computer games so popular among game players.

2.1.3 Using Computers for Learning by Game Playing

In 1978 at the University of Essex, UK, testing of network capacities led to the development of a text-based multi-user online game generally known as MUD (*Multi-User Dungeon*) [Bartle, 2003]. In contrast to PLATO, MUDs could be played on home computers and used the upcoming client-server networks. A typical MUD provided the user with an adventure scenario in a text-based virtual environment and allowed for written communication. This way of participating in a game led to a psychological effect called *imagined immersion* that had been unknown so far [Mitchell, 1995]. It also marked the beginning of a new era of virtual environments: collaborative learning environments and interactive 3D computer games.

MUDs remained popular among the limited circle of computer science students until the release of TinyMUD and MOO (*MUD Object-Oriented*) by the late 1980s [Bartle, 2003; Curtis and Nichols, 1993]. The MUD-derivatives facilitated the creation of virtual scenarios because they provided an object-oriented scripting language that could be handled by far more users. This resulted in numerous collaborative learning environ-

(a) (b)

Figure 2.4.: Screenshots from the *World of Warcraft* (WoW) game. A scene from the game (a). Users are represented by player characters, but the game additionally consists of non-player characters that are controlled by the game software. WoW also contains an authoring editor (b). This can be used to modify the virtual world and to create so-called *MODs*.

ments like *MOOSE Crossing* and many others [Bruckman, 1997; Fanderclai, 1995]. By the mid 1990s, improvements to graphics hardware made possible real-time interactive 3D graphics and paved the way for so-called MMORPGs (*Massively Multi-User Online Role Playing Games*). Games like *EverQuest* and *World of Warcraft* extended *imagined immersion* to *visual immersion* with ongoing success (see Fig. 2.4(a)).[4] Special authoring editors were introduced to let users modify existing games, which are then referred to as *MODs* (see Fig. 2.4(b)).

The huge success of these games prompted the U. S. Department of Defense in 1996 to deploy the 3D first-person shooter game *Doom II* for training purposes [Capps et al., 2001; Egenfeldt-Nielsen, 2005].[5] Soon thereafter the U. S. Department of Defense started to develop what is regarded as the first serious game—*America's Army* [Zyda et al., 2003]. The game is a tactical shooter to train military operations, even online, yet its main purpose has been to serve as a recruitment tool for the U. S. military (see Fig. 2.5).

4 For further information see [http://eqplayers.station.sony.com/] and [http://www.wow-europe.com/en/index.xml] (March 22, 2009).

5 The *Doom* series was created by Id Software. The U. S. Department of Defense developed *Marine Doom* in 1998 to teach teamwork, coordination, and decision making. *Marine Doom* is no longer used by the Marine Corps.
For further information see [http://www.dodgamecommunity.com/modules.php?op=modload&name=News&file=article&sid=31] (March 22, 2009).

(a) (b) (c)

Figure 2.5.: From *Doom* to *America's Army*. Screenshot from the first-person shooter game *Doom* (a). The user generally sees only his own hands holding the currently loaded weapon. *America's Army* was intended to train the U. S. soldiers (b). The screenshot illustrates the player's view of the scene. Another screenshot from *America's Army* shows the other players in the game (c). The graphics aim at photo-realism.

Serious games have been produced for several different markets since. Among others, the games target military, corporate, and educational application domains [Michael and Chen, 2006]. With this development, the research field of digital game-based learning emerged, which focuses on developing digital game-based training systems to improve learning and education [Gee, 2003; Prensky, 2001]. While early systems like PLATO applied computer technology to support conventional educational methods, programming environments like *Logo* and *Squeak Smalltalk* started to actually make use of the computer technology to support new ways of learning and understanding. Game elements were considered as add-ons (cf. Rieber [1996]). By now, digital game-based learning considers game elements themselves to be crucial to enhancing learning. Motivational and immersive potential of computer games is examined to better involve the learner in the game world.

As was mentioned in the introduction, the immersion impact of games can also lead to addictive behavior of players [Rehbein et al., 2009]. Also, the depiction of violence in computer games has been controversially discussed for many years. Several empirical studies have shown that violent computer games can increase human aggression [Schmidgall, 2006]. This increase may be due to the game's immersion depth and the user's interactive participation, in contrast to other media [Anderson, 2003]. Anderson [2000] therefore recommends creating computer games that enhance learning of valuable content. Before examining how digital game-based learning pursues this goal, the following section briefly reconsiders necessary terminology.

2.2 Terminology

When talking about computer applications for learning the terms *learning*, *training* and *education* are often used almost interchangeably. Yet a clear definition of these terms is required for developing a general framework for digital game-based training systems that suits education.

Learning: In psychology learning is considered an internal process that leads to behavior modifications [Bower and Hilgard, 1983; Bruns and Gajewski, 2002; Lefrancois, 2006]. This process has not been completely understood so far. Therefore, distinct theories have evolved to describe the learning process, i.e., to describe what has led to the behavior modifications. The main theories are *Behaviorism*, *Cognitivism*, and *Constructivism*:

- Behaviorists describe learning by relating behavior modifications to external stimuli. Learning itself is regarded as a "black box" [Skinner, 1974].
- Cognitivists describe learning as information processing. Behavior modifications are caused by thinking and reflection [Lefrancois, 2006].
- Constructivists describe learning as information construction that depends on individual experiences and personal hypotheses of the environment [Piaget, 2003]. *Constructionism*, which was mentioned in Section 2.1.1, can be regarded as Constructivist derivative.

To elicit the behavior modifications, different training strategies can be conducted based on the learning theories. Ertmer and Newby [1993] suggest using different training strategies for different training purposes. According to the authors, behaviorist strategies support training of facts and content, the *what* of education, whereas cognitivist strategies support understanding problems and training of problem-solving skills, the *how* of education. Constructivist (and constructionist) strategies are considered supportive for training of reflection on problems, the *why* of education (cf. Ally [2004]).

Training: In *The Oxford Dictionary of Current English* [2008] training is considered as actively teaching or being taught a certain skill or behavior. Marshall [1998] describes it as a preparation for a specific task by ordered instruction that implies instruction as well as exercising. Training can be understood in the sense of "to train something" to support one's own learning process as well as "to train someone" to support someone else's learning process. Thus, it can be considered an active process of exercising that follows some kind of instruction to support learning.

Education: According to *The Columbia Encyclopedia* [2007] education is either an informal or a formal process of shaping an individual's potential. Informal education is a subtle process caused by the environment in which an individual acts. Formal education on the contrary is a conscious process to impart skills and knowledge which are considered to be necessary in order to live in a given society. Formal education can be regarded as formal environment for learning that gives room for exercising and instruction.

Following the definitions, learning is an individual process that can be supported by different training strategies and is shaped by education. In regard to software applications, numerous software types have been developed based on the different learning theories [Baumgartner and Payr, 1999]. Distinct software applications provide learners with training strategies, yet rarely address the broader field of education as application domain:

Drill & practice systems: The applications typically pursue Behaviorist strategies to reinforce already learned concepts [Baumgartner and Payr, 1999]. They offer linear sequences of tasks to practice skills. For example, PLATO uses drill & practice strategies. If accompanied with game or multimedia elements, drill & practice systems are generally referred to as *edutainment* applications.[6]

Computer games: The applications typically pursue Cognitivist strategies (cf. Baumgartner and Payr [1999]). According to Crawford [1984], a game is a subjective and artificial representation of a real situation that is based on a conflict or problem. The conflict is presented in a game context, that is the game world and story. Rules determine how to play the game and how to solve the conflict. Levels structure the game. Game playing is generally non-deterministic, that is different strategies can lead to the same goal. Hence the player has to learn how to solve problems. **Serious games** denote any kind of game that serves an educational purpose in a more authentic way than traditional computer games. The learning objectives are directly incorporated into the game [Michael and Chen, 2006].

6 Edutainment is the abbreviation for *educational entertainment*. Meanwhile, edutainment applications have gained a bad reputation because the entertaining elements often are not related to the learning objectives [Dondlinger, 2007; Egenfeldt-Nielsen, 2006].

MicroWorlds and MODs: MicroWorlds naturally pursue Constructivist/Constructionist strategies. The environments allow the user to interactively reconstruct content and by this support understanding of how and of why things work [Baumgartner and Payr, 1999; Rieber, 1996]. A MOD is a game-based version of a MicroWorld [El-Nasr and Smith, 2006]. Commercial computer games[7] are often shipped with an authoring editor that allows the user to modify and enhance the game world, rules, behavior, and characters (cf. Sec. 2.1.2). This kind of authoring is called *modding* and can result in a new game — which is called a MOD.

The examples demonstrate how the different learning theories and training strategies have been applied to software applications. Different technologies can be used to implement such software applications. As was already mentioned, virtual environments have long been used as a basis for collaborative learning environments as well as for computer games. In this dissertation, the focal point will be on interactive 3D environments for desktop PCs:

Interactive 3D environments: This type of virtual environment is a 3D representation of a virtual world or scene, also known as *3D virtual environment*. Scene objects are represented as interactive 3D graphics. Based on the functionality added to the environment, users can explore and modify the scene. Moreover, users can be represented in the interactive 3D environment as virtual characters.

Given that the interactive 3D environment implements learning objectives, it is generally referred to as an *interactive 3D learning* or *training environment*. A digital game-based training system will furthermore be defined here as:

Digital game-based training system: An interactive 3D training environment that uses computer game design concepts like game conflicts, stories, rules, and levels to include, structure, and present learning objectives.

Having reconsidered necessary terms and definitions, the next section is going to introduce the state of the art of digital game-based learning and digital game-based training systems.

7 Commercial computer games are also referred to as *COTS computer games*; COTS is the abbreviation for *commercial off-the-shelf*.

2.3 State of the Art

As van Eck [2006] points out, the sales figures of computer games speak for themselves when thinking of using game technology for interactive 3D learning and training environments. Yet what makes a game good? Pausch et al. [1994] clearly identify the uniqueness of computer games by stating

> "Computer users have tasks they need to perform, and are therefore motivated to overcome poorly designed interfaces. With video games, there is no external motivation for the task — if the game's interface is not compelling and entertaining, the product fails the marketplace." [Pausch et al., 1994, p. 177]

Central to this statement is that users *do not have to* play computer games. They do it for entertainment reasons — given an engaging game.

Compelling interface design is one aspect that makes a game good. Apart from that, researchers have tried to identify other features that make games good. This is reflected in several definitions (cf. Salen and Zimmerman [2004, Chap. 7]). According to Salen and Zimmerman, Crawford [1984] is the first to define games from a computer science perspective:

> "A game is a closed formal system that subjectively represents a subset of reality. By "closed" I mean that the game is complete and self-sufficient as a structure. ... By formal I mean that the game has explicit rules ... " [Crawford, 1984, pp. 7/8]

Crawford [1984] points out three further properties of computer games: interaction, conflict, and safety of experience, that is, playing without having to fear physical restrictions. Salen and Zimmerman [2004] conclude that

> "A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome." [Salen and Zimmerman, 2004, p. 96]

Both definitions focus on the technical aspect of game design. However, components like artificial world, contest, user interaction, and outcome could also be part of any simulation or learning environment [Champion, 2005]. Champion therefore defines a game as follows:

> "A game is a challenge that offers up the possibility of temporary or permanent tactical resolution without harmful outcomes to the real world situation of the participant." [Champion, 2005, p. 41]

Champion introduces the terms *challenge* and *resolution*. He claims the combination of both to be the key to success of good games, and, hence, the key to keeping players motivated to continue playing. The author further outlines that good games achieve this combination by challenging the player with riddles and obstacles that are manageable and regularly resolved with rewards (cf. also Gee [2003]). In addition, Champion points out *how* game playing implicitly trains the players: with each game level the difficulty to resolve the riddles increases. As a consequence, players have to enhance their (game-dependent) skills to proceed (cf. Pivec et al. [2003]).[8] An interactive training environment that causes similar enthusiasm would be desirable and is target to the research field of digital game-based learning.

The remainder of this section is structured as follows: the research field of digital game-based learning is briefly outlined first (cf. Sec. 2.3.1). Next, recent digital game-based training systems are discussed in Sections 2.3.2 and 2.3.3. Subsequently, different evaluation studies, in particular one study by Egenfeldt-Nielsen [2005], will be closely examined that elucidate *why* digital game-based training systems have not become well-received educational tools so far (cf. Sec. 2.3.4). A discussion will finally conclude the state of the art (cf. Sec. 2.3.5).

2.3.1 Digital Game-Based Learning

The term *digital game-based learning* was coined by Prensky [2001] in his equally titled book [Prensky, 2003]. The author strongly promotes use of computer games for learning because, as Prensky points out, today's students and teachers speak different languages. At least, teachers are generally assumed to be unfamiliar with computer games (cf. also Martens et al. [2008]). Prensky believes that it is learning itself what motivates players to continue playing. In addition to obvious game skills like flying an aircraft or playing tennis, he argues, computer games support to train deeper skills like

8 For instance, a first-person shooter game like, e.g., *Doom* requires the player to know how to handle a weapon and to shoot his enemies instantly whereas a strategy game like, e.g., *Civilization III* requires the player to carefully plan in advance before making decisions to win in the long run.

coping with different information sources, making decisions, developing strategies, and understanding complex systems [Prensky, 2003, p. 3].

Gee [2003] finds games especially suitable for learning because the applications provide instant feedback and context information throughout the game (cf. Greitzer et al. [2007]). Consequently, problems could be dealt with more appropriately and information could be processed according to personal learning progress. Like Champion [2005], Gee states motivation for playing to be fostered by challenge and resolution. Yet in addition, Gee [2003] considers slipping into game roles and role identification to be what fosters reflection on and interaction with the context. Besides, computer games seem to be the crucial medium to reach young learners these days [Jenkins et al., 2003].

The following two sections discuss different approaches to implement digital game-based training systems. Starting with the traditional approach that sifts individual game design concepts to interactive 3D learning and training environments, the subsequent section draws on with a description of serious games.

2.3.2 Learning and Training Environments

As was just outlined, early approaches of digital game-based learning added individual game design concepts to interactive 3D learning and training environments. These mainly focused on game-based interaction and interface design as well as on game-based rendering styles and the integration of game roles. Game-based interface design is generally considered to be helpful to acquaint users with software applications [Korten et al., 2002; Malone, 1982; Maxwell et al., 2004; Pausch et al., 1994].

AquaMOOSE 3D Elliott and Bruckman [2002] present *AquaMOOSE 3D*, a game-like environment for maths education. Initially, *AquaMOOSE 3D* was inspired by the MUD *MOOSE Crossing* (cf. Sec. 2.1.3) and, thus, was intended as a multi-user environment for communicating on maths. However, due to technical problems it was limited to single-user access; it only addresses students as game players. What is most interesting about *AquaMOOSE 3D* is the adaptation of game elements to learning objectives. The game elements concern the virtual environment's theme as well as the creation of visual appealing mathematical artifacts. Learning objectives are to engage in parametric equations following Papert's constructionist approach [Elliot, 2005]. To do so, the virtual world is situated in an underwater scenario. The user is represented by a fish avatar

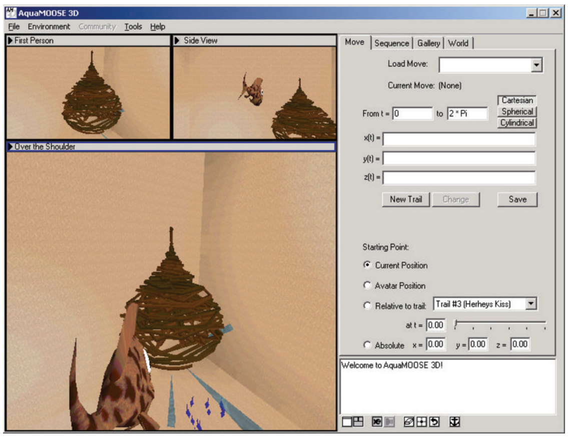


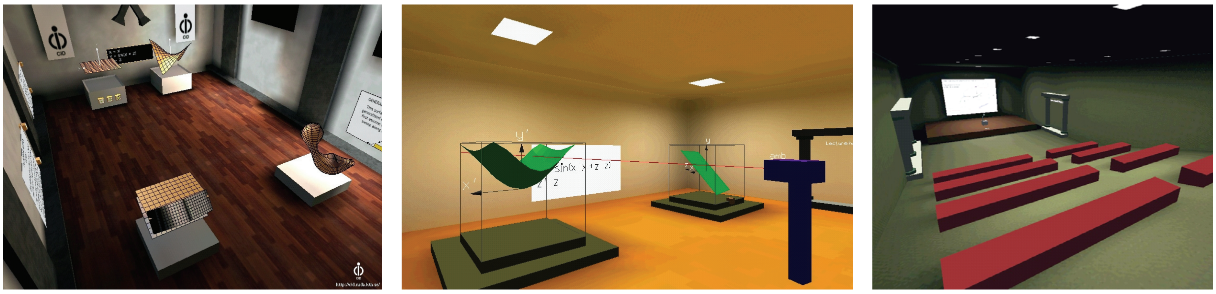
Figure 2.6.: *AquaMOOSE 3D* is an interactive graphic 3D environment for engaging with parametric equations by creating artificial objects. The entered equations are displayed as trails in the environment and can be traversed by the fish avatar [Elliot, 2005].

(see Fig. 2.6). Learning about parametric equations is achieved by implication: students have to determine the fish's movement by modifying and solving equations. Since fishes can swim around freely and leave trails in the water they provide a visual and practical representation of the abstract maths theories. *AquaMOOSE 3D* gives a good example of relating abstract theory (maths) to a different domain (underwater world) for explanatory purposes.

CyberMath In contrast to *AquaMOOSE 3D*, *CyberMath* is a maths environment that aims at integration into education [Taxen and Naeve, 2001]. *CyberMath* is an avatar-based and collaborative 3D virtual environment to teach mathematics. The authors point out that an easy integration into school environments has been provided by implementing a desktop-VR-application. Users can explore the environment alone or in groups by applying a game-based navigation control, was chosen due to its popularity among students (see Fig. 2.7). Furthermore, the system provides a laser pointer tool for visual communication (see Fig. 2.7(b)).

Interestingly, guided exploration and lecture presentation facilities are integrated into the environment to address teachers. The authors believe that teacher participation strengthens the system's support to learning (see Fig. 2.7(c)). Since the system targets two user groups, visual communication functionality, and curriculum integration, it would be of great interest to know about the utilization at school. However, no further publications present evidence on this point.

Aircraft Maintenance Training Apart from maths environments, other training fields have been targeted. Hintze et al. [2001] present a flexible and interactive training environment for aircraft maintenance training (cf. also Bluemel et al. [2003]). The



(a) (b) (c)

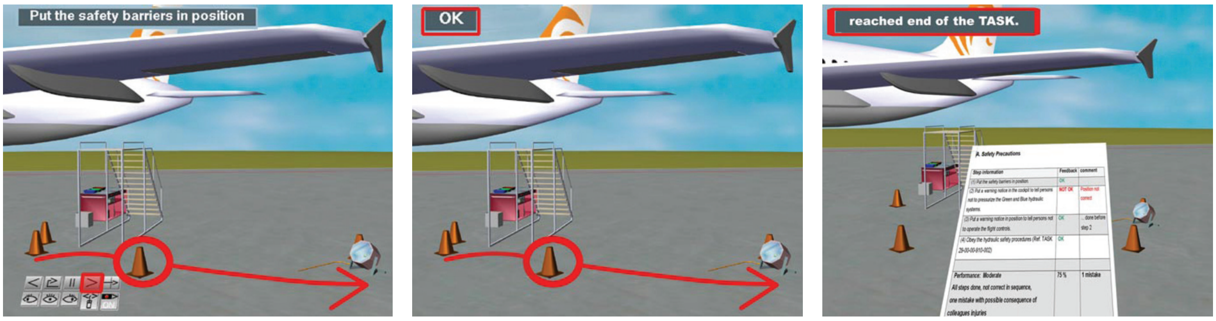
Figure 2.7.: Screenshots from *CyberMath* [Taxen and Naeve, 2001]. The exploratory site shows different mathematical exhibitions (a). The red line denotes the laser pointer by which users can point at and manipulate objects (b). The lecture hall can be used to present lectures (c).

system provides different interaction modes that allow the user to interact with the environment and to accomplish training tasks in a game-based point-and-click manner. Although the system does not target higher education, it was chosen to demonstrate design of task-oriented training environments that provides authoring functionality.

In addition to support aircraft maintenance training, the system targets authoring of training scenarios. Therefore, an author has to establish state-based training tasks by defining scenario objects and their interdependencies, that is, their causalities. A scenario object, which is usually linked to a scene object, contains certain properties that define the current system state. A criterion is a certain state-set the user has to meet by modifying the scene object's properties. Whenever a criterion is met by the user an action is triggered by an according causality, which combines criterion and action. For example, if the user has accomplished a task (met a certain criterion) he can proceed with the training session (an action triggers the next task). Bluemel et al. exemplify the environment with an aircraft maintenance scenario as illustrated in Figure 2.8.

The state-based task planning approach employed by Bluemel et al. [2003] is a standard approach for many training environments (cf. [Haller et al., 2000]). It is based on the underlying implementation that uses scene graphs. In regard to authoring and training, Bluemel et al. [2003] emphasize the cost effectiveness of interactive 3D environments for maintenance and training experience (cf. de Oliveira et al. [1999]). Incorporated expert knowledge would reduce the amount of teacher time necessary [Bluemel et al., 2003].

Reduction of teacher time is a typical goal pursued by many digital game-based training systems. As will be discussed in Section 2.3.3, most applications only address students



(a) (b) (c)

Figure 2.8.: Screenshots of an aircraft maintenance scenario [Bluemel et al., 2003]. In presentation mode the tasks are highlighted for the user. On the lower left side a menu is displayed which the user can use to step through the individual task stages (a). In guided mode the user has to interact with the objects and accomplish the tasks (b). The system highlights what to do. Additional help is not displayed automatically in free or discovery mode. The user can ask for it, though. The author's scenario view provides an editor to create scenario objects and to link them to goals (c).

as players and neglect the role of teachers in education. In that sense, *CyberMath* can be considered an exceptional interactive 3D training environment.

Further Examples Apart from the systems presented in the previous paragraphs, numerous other learning environments have been developed. Most target maths-related topics like interactively learning how to program [Kay, 2005], 3D computer graphics [Howard et al., 1999; Petterson et al., 2003], or interacting with 3D objects [Benford et al., 2000]. Corradini et al. [2005] implement an interactive conversational character that impersonates the fairy tail writer Hans Christian Andersen. Users can communicate with the character to learn about his fairy tails. Online learning with game elements is investigated by Bouras et al. [2004].

The 3D virtual online environment *Second Life*® also implements game elements and allows for training. Its use in education has been surveyed only recently [NMC, 2007].[9] The results point towards an increase of using 3D learning environments for education.

Apart from the development of digital game-based training systems, digital game-based learning also pursues the idea of supporting learning and education with game design itself. Therefore, MODs and game authoring editors are used [El-Nasr and Smith, 2006]. Robertson and Good [2005] and Szilas et al. [2007] investigate story creation to support teamwork skills and logical thinking. Korte et al. [2007] teach theoretical computer sciences using game development for student's assignments.

9 For further information see [http://secondlife.com] (March 22, 2009).

The last subsections presented interactive learning and training environments that use individual game design concepts like game worlds and game-based interaction techniques to enhance the applications. Serious games take one step further, as will be discussed below.

2.3.3 Serious Games

Section 2.3 examined different definitions to describe games. Most decisive aspects are: games define a closed context, the game context uses rules to structure how players can proceed from level to level, levels increase in difficulty, games establish stories, games offer players to slip into roles, games challenge and reward players with specific game-tools, an so on. In contrast, most traditional interactive 3D learning and training environments implement learning goals using game elements, but do not provide a game-like system.

Serious games denote games that pursue to educate rather than to entertain. Like computer games, serious games have an industry background. Several different categories of commercial serious games exist; educational games being one among them. For instance, *Stone City* trains employees of the Cold Stone Creamery ice cream company how to serve customers.[10] *Food Force* has been developed for the United Nations and is about how to prevent famine at the fictional island of Sheylan.[11] *Power Politics III* simulates U. S. presidential elections back to the 1960s and targets winning the presidency.[12] The serious game development platform *OLIVE* (*On-Line Interactive Virtual Environment*) allows to create a range of applications from business to military to governmental contexts with an included an authoring editor.[13] Figure 2.9 illustrates screenshots of the presented games here.

Numerous other serious games exist and sales figures are increasing [Susi et al., 2007]. Research has become aware of this development and started to investigate [Capps, 2002; Raybourn and Bos, 2005]. The subsequent examples are of special interest for

10 For further information see [http://www.persuasivegames.com/games/game.aspx?game=coldstone] (March 22, 2009).

11 For further information see [http://www.food-force.com/] (March 22, 2009).

12 For further information see [http://www.powerpolitics.us/home.htm] (March 22, 2009).

13 For further information see [http://www.forterrainc.com/] (March 22, 2009).

(a) (b) (c)

Figure 2.9.: Screenshots of commercial serious games. Rescuing and managing crisis situations is one major goal of the *OLIVE* serious game environment (a). The visualization, thus, is done very realistically. The Cold Stone Creamery company employs a more artificial rendering style to train their employees (b). The screenshot shows an ice cream shop with one employee serving customers. Power Politics III also employs realistic visualizations (c). However, the serious game does not need to be visualized in a virtual 3D environment all the time. Rather the game applies well-known depictions of president races.

this dissertation because they help to identify strengths and weaknesses of current approaches.

Levee Patroller Harteveld and Bidarra [2007] present the levee inspection game *Levee Patroller* for levee patroller workshops. Technically, the game trains emergencies caused by floods. The player can practice how to recognize levee failure symptoms and how to communicate these to the field office in a first-person shooter manner. The game uses the commercial 3D game engine *Unreal Engine 2 Runtime* and is denoted as a *conversion MOD* by the authors. See Figure 2.10 for two screenshots of the serious game.

What is most interesting about *Levee Patroller* is that it was developed in collaboration with expert patrollers and that it was evaluated at a patroller's workshop. Three main findings of the evaluation depict the current research situation in regard of using serious games for education. According to Harteveld and Bidarra [2007, Chap. 4] "games need to be a single whole," "games function as discussion-support-tools," and "games request a complementary learning style." With the first finding the authors state that games

Figure 2.10.: Screenshots of Levee Patroller [Harteveld and Bidarra, 2007]. The images show two destroyed parts of a levee caused by a flood.

can only depict one specific area of interest. They had been faced with several different opinions on levee patrolling by the collaborating experts and concluded that a game could only incorporate few aspects. Harteveld and Bidarra propose enhancing authoring tools to support distinct scenarios and to better integrate games into workshop curricula. The second finding they claimed to be a result of the first. The limited game world triggered active discussions during the workshop about whether the game scenario was realistic or not. Therefore, the authors call for communication support. The third finding points towards the problem of suitability. The authors state that games are best at motivating users when they are linked to reality, normally through discussions and explicit links to real training procedures.

In addition to these results, Harteveld and Bidarra suggest that a change in the curriculum is required if games are to be used as supplement to education. They argue that discussions would be primarily fostered by the game and that these required time. Similar conclusions have also been drawn by other researchers [e.g. Egenfeldt-Nielsen, 2005; van Eck, 2006; Warmerdam et al., 2007]. This is true for practical training as well as for higher education.

SGTAI BinSubaih et al. [2008] introduce *SGTAI*. The game is developed for traffic accident investigators, especially novices, in Dubai and targets an application field similar to this dissertation. The serious game has been developed to provide the investigators with a personal training tool to interactively train scene assessment. Current training is divided into lessons and "on-the-job" experience only. The investigators have few possibilities to recapitulate scene assessment procedures. BinSubaih et al. further argue that SGTAI could provide a large variety of training cases and, thus, better support novice and expert investigators.

SGTAI was planned as a first-person shooter game. Five investigative phases were implemented: receiving the incoming call, arriving at the accident scene, conducting the initial investigation, finalizing the data collection, and completing the accident file. To evaluate the user's performance and learning progress, SGTAI has an integrated multiple-choice-like self-evaluation wizard that results in a score sheet. Additionally, the application logs the user interactions and movements to generate a textual and visual log file (see Fig. 2.11(c)). The game was implemented with *Torque* engine, a commercial 3D game engine. It should allow for exploration and interaction with the environment, although this was not explicitly described in the paper. Figure 2.11 illustrates screenshots of the final game.

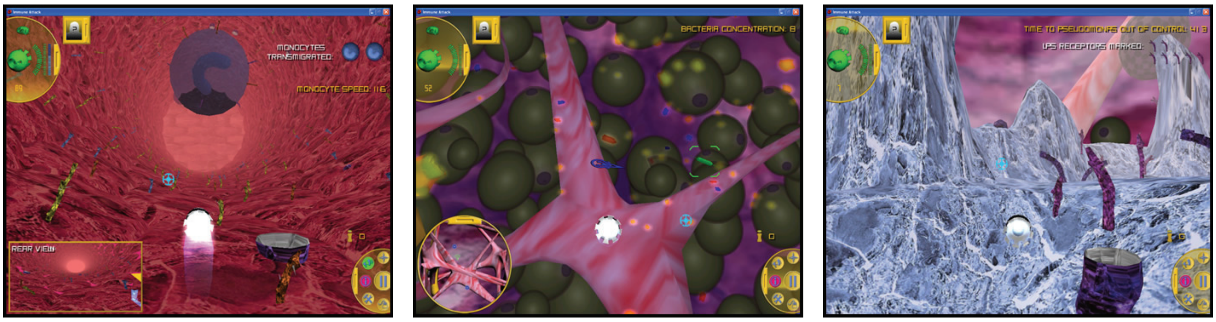
(a) (b) (c)

Figure 2.11.: Screenshots of *SGTAI* [BinSubaih et al., 2008]. The traffic accident scene during scene assessment. The investigators have to train how to protect the scene (a). Among others they have to label clues and issues (b). One part of *SGTAIS*'s training feedback is the navigational patterns overview (c). This allows to review the movements done by the investigators during scene assessment. These could be used to discuss effectiveness, wrong behavior, prioritizing, etc.

Most interestingly, the authors conducted an evaluation study to prove the effectiveness of SGTAI for learning. The study employed a control group A that did not use SGTAI and a experimental group B that played the game. Both consisted of novice and expert traffic investigators. A pre- and a post-test were used to verify performance differences of the groups before and after using SGTAI. Two hypotheses were to be proved: first, the use of SGTAI improves group B's performance compared to group A; second, novices improve their performance more compared to experts using SGTAI. The first hypothesis could be verified statistically, whereas the second could be verified only to a lesser extent.

The authors provide three arguments for the first result: better performance caused by the game was due to concentration and interactive participation which fostered the learning process. Better performance was a result of the integrated high score which challenged and motivated the users to achieve better results. And better performance was caused by achievable goals which kept the players motivated throughout the game. The second result is discussed less detailed. The authors acknowledge significant differences in the improvements of novices and experts and suppose this might be due to experienced investigators adopt shortcuts when assessing the scene. Hence, they do not run through the whole training session. BinSubaih et al. conclude that SGTAI could also be useful to refresh expert's knowledge. In addition to these results, the authors stress that the game's realism was well received by the players. This goes along the lines of Harteveld and Bidarra [2007] concerning the acceptance of serious games' authenticity.

Immune Attack Kelly et al. [2007] present a serious game called *Immune Attack* for teaching immunology to high school and college students. The publication is an outstand-



(a) (b) (c)

Figure 2.12.: Screenshots of *Immune Attack* taken from the project's website. The picture shows the microscopic robot within the blood vessel (a). The player has to train an immune cell to transmigrate out of the blood vessel into the connective tissue. Here, the player has to train the immune cell to follow the right trail to the infection (b). Finally, the player has to train the immune cell how to recognize the bacteria outside of the blood vessel (c).

ing example for the design of a serious game in terms of collaboration between experts, immunologists, game developers, and teachers, which resulted in close adaptation of the game world to the learning goals.

The game is situated in a virtual body through which the player navigates in a microscopic robot. The mission is to activate the body's immune cells so that they fight an infection. The player is regularly acquainted with facts in order to proceed from level to level and implicitly gets acquainted with the immune system's inner processes. In the first level the player navigates through a blood vessel and is informed that an infection is nearby. The player receives hints and has to train a macrophage, a basic immune cell, to do several tasks like finding the blood vessel closest to the infection, exiting the vessel, following a chemical trail that leads to the bacteria creating the infection, and finally killing the bacteria without destroying healthy cells (see also Fig. 2.12). As the levels increase, the cells and bacteria become more and more sophisticated and the player has to access context dependent help in order to proceed.

During game design the authors had to meet two daunting challenges: first, how to integrate help tools and guides; second, how to visualize and simulate biological processes correctly at different physical and time scales. To meet the first challenge the authors employed *My Learning Assistant* (*My LA*), an interactive question asking-and-answering tool. *My LA* provides guided cooperative questioning and uses game context information to answer the players questions [Patil and Howell, 2007]. The second challenge could not always be met. The game developers had to simplify several physical scales, e.g., the amount of blood vessels had to be reduced. Nevertheless, a conducted evaluation study responded positive results. Especially teachers were strong supporters of new teaching methods. The game is currently undergoing finishing touches and will be released in the near future.[14]

Serious games still have not become a well-received educational tool. This might be because of prevailing prejudices towards game use in education but also because game-like software applications still lack educational potential. The following Section 2.3.4 closely examines a study on game use in class that gives insight into the difficult relationship between education and games.

2.3.4 Evaluating Games in Education

Several studies evaluated the effectiveness of computer games for learning [Clarke and Duimering, 2006; Swing and Anderson, 2008; Wong et al., 2007]. However, only few studies evaluated the effectiveness of actually using computer games in education [Egenfeldt-Nielsen, 2005; Squire, 2004]. Here, the study of Egenfeldt-Nielsen [2005] will be in focus. It was conducted over a fixed period of 2 ½ months at a Danish high-school and involved 72 students and 2 teachers. The results conform to general expectations regarding motivational potential of computer games but also yield unexpected difficulties.

Egenfeldt-Nielsen took up the study of Squire [2004] who had proved the effects on motivation when playing the history strategy game *Civilization III* in class. Egenfeldt-Nielsen believed that playing and discussing a history computer game in class would foster reflecting on the game and linking game experience with textbook materials. His study targeted two central points: first, do computer games fit current school settings? Second, how do students perceive and play computer games and do they possibly learn through computer games?

Two experimental groups and one control group were selected. The experimental groups had to play the history strategy game *Europa Universalis II* whereas the control group had to work with case studies. The participating teachers had to give the same lecture and textbook materials to all groups throughout the study. Teaching the experimental group then varied between playing the game, reflecting on the game, and lessons. Teaching the control group varied between group work on the case studies, discussing the case studies, and lessons. To verify the learning progress of the participants, the groups had

14 For further information see [http://fasweb.beacontec.com/immuneattack/] (March 22, 2009).

to pass a factual test after one month and a factual retention test after five months of the evaluation study. Furthermore, the author prepared questionnaires and interviewed the participants.

Results of the Study

Results were mixed. Egenfeldt-Nielsen detected the expected motivation boost at the beginning of the study and additionally found two unexpected but insightful results. First and most surprisingly, the students were far more positively prejudiced towards using computer games for teaching history than were the teachers. The students actively reflected on and questioned the educational use of computer games throughout the study. They rarely used the computer game as an educational tool and did not trust the computer game's presentation of historical facts. Egenfeldt-Nielsen suggests this was caused by the history topic and because the computer game slightly changed facts in favor of a subjective and entertaining view on history. This was steadily criticized by the students.

The second problem was that teachers were foreign to computer games and could not easily intervene or instruct the students during playing. This seems to be a well-known problem [Jenkins et al., 2005; NMC, 2007]. Moreover, the teachers were not able to adapt their teaching methods to the new media. Using computer games in class required time for discussions of student's interpretations. This was generally rejected by the teachers due to time constraints. It conforms to an evaluation study on using 2D virtual learning environments for higher education by Jenkins et al. [2005].

The statistical results of Egenfeldt-Nielsen's study indicated higher intrinsic motivation for the experimental groups compared to the control group. This conforms to the majority of studies conducted in the field [BinSubaih et al., 2008; Clarke and Duimering, 2006; Harteveld and Bidarra, 2007; Squire, 2004]. Egenfeldt-Nielsen's study did not exhibit remarkable differences in the learning outcome of the experimental groups compared to the control group. This had also been the result of an evaluation concerning *AquaMOOSE 3D*, but was explained due to poor graphics and few exiting game elements [Elliot et al., 2002]. In contrast, Wong et al. [2007] found positive effects on the learning outcome when evaluating different multimedia tools compared to textbook materials. Also, BinSubaih et al. [2008] found positive learning outcome after evaluating the usage of *SGTAI*. Hence, common ground is not found for this particular issue.

Egenfeldt-Nielsen [2005, Chap. 8, pp. 225]concluded that "students and teachers never really got the idea" and stressed that the tools to measure the learning outcome were not strong enough. He insisted on the effectiveness of computer games for learning and called for more suitable applications. A crucial challenge, Egenfeldt-Nielsen argued, was to balance engagement of playing and engagement of learning. The author admitted that playing will be favored over learning if instruction is missing and the game is too entertaining. Students then were not able to connect both aspects.

2.3.5 Discussion

Digital game-based learning is a relatively young research field that examines computer games in order to adapt them to education. Until recently, research focused on adding single game elements to interactive 3D training environments. By now, research started to investigate serious game design to improve digital game-based training systems. Serious games have a great potential for improving learning situations; recent studies substantiated that the applications motivate students and cause lively classroom discussions. However, new forms of instruction as well as discussion are required for integrating serious games into education [Egenfeldt-Nielsen, 2005; Harteveld and Bidarra, 2007]. Teachers have to get used to this kind of technology; students have to get used to comparing and reflecting on information sources, for instance. Moreover, serious games need to be more authentic if they are to be considered as an educational tool by students.

Yet there are more challenges that have to be faced by research. Compared to non-digital games that have long been used in education, serious games are considerably more restrictive: the applications usually present a fixed set of rules, content, and rewards that can not be adapted to the lessons. Teachers are neither addressed nor supported by the applications; instead, serious games tend to replace teachers and conventional teaching methods. This is unlikely to be successful because it reduces education to a question–and–answer game.

Digital game-based learning has to face these challenges in order to fully exploit the potential of serious games for education. The applications could serve well as a learning motivator and training sessions organizer, for instance. Therefore, serious games have be made suitable for education first: teachers as well as students have to be addressed as users and have to be acquainted with the software applications. At the same time, serious game's potential to support classroom discussion has to be further exploited. To do so, training session review logs could be generated and used as a basis for classroom discussion.

In conclusion, a general framework for digital game-based training systems is required that addresses all of these issues and provides a basic implementation pattern for future applications. The development of such a general framework is the central goal of this dissertation and will be described in Chapter 3. As a basis for the framework, educational situations need to be examined. Here, police education — in particular, crime scene investigation (CSI) training — serves as the application scenario for developing a general framework for digital game-based training systems. CSI training is well-suited because it combines practical and theoretical training aspects. The next section describes common educational situations of CSI training.

2.4 Application Scenario

The area of application employed in this paper is CSI training. Expertise has been provided by the Police College of Saxony–Anhalt, Germany, in the course of a collaboration project with the Computer Science Department of the Otto–von–Guericke University of Magdeburg, Germany.[15] The project targeted the development of a digital game-based training system for CSI to be deployed at the Police College. The University of Magdeburg was responsible for the design and implementation of the prototype whereas the Police College provided insight information on the teaching methods and the curriculum of studies. The following subsections will therefore introduce education at the Police College as well as CSI training. In-depth details will not be presented for reasons of confidentiality.

2.4.1 Police Education in Saxony–Anhalt

In Germany, police education is a domain of the state government. Each federal state organizes a Police College. Slight differences in curriculum, choice of candidates, or provided qualification may therefore occur. The Police College of Saxony–Anhalt undertakes education of traineés for middle and upper grade of civil servants in the police force. Courses last 2 ½ and 3 years each and consist of theoretical knowledge and practical professional experience. Lessons are divided into theoretical teaching and practical training. Teachers at the Police College have been working professionally in the police

15 For further information see [http://www.polizei.sachsen-anhalt.de/index.php?id=174] and [http://wwwisg.cs.uni-magdeburg.de/] (March 22, 2009).

(a) (b)

Figure 2.13.: Crime scenes for training at the Police College of Saxony–Anhalt. The exterior crime scene for training (a). The interior crime scene for training with simulated burglary (b). Images courtesy of the Police College Saxony–Anhalt, Germany.

forces or other fields of law enforcement. Thus, on-the-job expertise is offered by the College.

The didactic body of the curriculum is organized by four departments *Leadership and Management* (Dept. I), *Criminalistic Sciences* (Dept. II), *Law* (Dept. III), and *Social Sciences* (Dept. IV). Students are broadly trained in police management, arms and ammunition, criminalistics, criminal law, also in psychology, sociology, politics, and ethics. Training of CSI, which is of concern for this dissertation, belongs to the criminalistic sciences. To suit training to real investigation, the Police College possesses an outdoor as well as an indoor crime scene for training (see Figs. 2.13(a) and 2.13(b)). A crime laboratory (crime lab) is used to train treatment of physical evidence.

One problem of current education at the Police College, however, is the separation of subjects. Distinct subject areas seldom relate topics to one another. Consequently, CSI training rarely contains questions that concern legal issues, for instance. A 3D virtual training environment could overcome this missing link.

2.4.2 Crime Scene Investigation (CSI)

Criminalistic sciences generally deal with revealing and resolving crimes. One part of criminalistic sciences is *criminalistic tactics*. Criminalistic tactics are specific procedures conducted by the police forces when a crime has been committed. Crime scene investigation (in German “Tatortarbeit”) is one of the first steps to be undertaken in the sequence of criminalistic tactics. It is concerned with assessing and processing the crime scene to reconstruct the crime. During CSI, the investigators have to apply *criminalistic techniques* to preserve and collect evidence. This is mandatory for resolving the crime and for using evidence in court.

The goal of the Police College is to train the students in conducting appropriate steps of criminalistic tactics on the one hand and to appropriately apply criminalistic techniques on the other hand. The following description only presents an excerpt of crime scene investigation tactics that have been selected for the application scenario of this dissertation. For further reading confer, e.g., Ashcroft et al. [2004]; Clages [1997].

Criminalistic Tactics

Criminalistic tactics are goal-oriented techniques used to organize the appropriate protection and assessment of the crime scene. The first action to be undertaken is called *initial response* (in German "Erster Angriff") that is divided further into *scene protection* (conforms to the German "Sicherungsangriff") and *scene assessment* (conforms to the German "Auswertungsangriff").[16]

Initial Response

The German Ordinance on Police Services defines initial response as follows: "Beim ersten Angriff sind alle unaufschiebbaren Feststellungen und Maßnahmen zur Aufklärung einer Straftat zu treffen" [PDV, 1985]. This means that all findings and measures inevitable to clear up the crime are to be taken without delay. Immediate measures include protecting the crime scene, treating injured, prosecuting suspects, interviewing witnesses, etc. Initial response is prompted by receipt of information that a crime has been committed. As mentioned above, initial response is divided into two phases:

Scene protection is the first phase of initial response. Central elements of scene protection are:

- receiving information; approaching the crime scene

16 It has to be noted, that the German division of *initial response* into *scene protection* and *scene assessment* does not conform with American terminology (cf. Ashcroft et al. [2004]). The German *scene protection* phase is equivalent with the American phase of *initial response*. Moreover, the German *scene assessment* is equivalent with the American phases of *scene assessment* and *processing of the scene*, conferring to [Ashcroft et al., 2004]. The elements and sequences of actions during all of the phases are equivalent, however.

- safety and emergency care; calling in other forces like fire fighters, medical doctors, etc.
- controlling people at the scene; arresting suspects; chasing fugitives; interviewing witnesses
- protecting and sealing off the crime scene
- protecting evidence and traces; in particular, that of transient nature
- recording actions and observations at the scene.

All through the investigation it is essential to avoid contaminating the crime scene. To do so, officers have to establish special pathways when assessing the scene. Also, they have to record any action carried out during the investigation. Recording is crucial when it comes to crime reconstruction and courtroom discussion.

Scene assessment is the second phase of initial response. It comprises assessing and processing the scene. Especially, collection and recording of evidence for crime reconstruction are central to scene assessment. Scene assessment procedures are

- conducting a first walk-through; establishing a pathway and minimizing contamination of the scene
- conducting an initial recording of the crime scene, evidence, and preliminary assumptions
- ensuring contamination control and identifying personnel in charge
- documenting the crime scene; coordinating photography, sketching, videotaping, etc., of the scene
- coordinating the collection of evidence, i.e., coordinating appropriate sequences of evidence collection (collecting transient and physical evidence first, for instance)
- collecting, preserving, and submitting evidence; observing the so-called *chain of custody*
- performing a final survey of the crime scene
- filing the case; collecting all kinds of documents, e.g., initial observations, recordings, photography, sketches, etc.

Specific preservation and collection procedures during scene assessment are part of the criminalistic procedures introduced in the next paragraphs.

Criminalistic Techniques

Criminalistic techniques are part of crime scene management. They are employed to gather evidence without destroying or altering it. Here, techniques for the collection of physical and documentary evidence are of special interest. As with criminalistic tactics, investigators have to apply criminalistic techniques appropriately and rationally. For instance, fingerprints have to be searched for at door knobs first, rather than on the ground. The subsequent descriptions of evidence and criminalistic techniques have been restricted due to confidentiality reasons.

Physical Evidence

This is regarded as visible or latent alteration of material found at the crime scene. Alteration can be caused by contact with hands (fingerprints), shoes (footprints), textiles (fibers), application of tools (tool marks), etc. The sequence of collecting, preserving, and submitting physical evidence is called *chain of custody*. Tools used during the chain of custody are called *evidence identifiers*. It is of particular importance not to destroy or alter evidence because it might lose its value. For the application scenario here, evidence types have been restricted to fingerprints, footprints, blood traces, and tool marks.

Fingerprints are dactyloscopic evidence. A fingerprint is determined by the epidermal (skin) ridges that are unique for every individual. Therefore, fingerprints are highly potential to identify humans. Fingerprints are generally latent. The traditional process of collecting fingerprints is called *dusting*: the officer has to brush the surface area the print is on with specific powder (graphite, lead, magnetic, ...), tape the print, and submit it. Choice of powders and brushes depends on the surface material the fingerprint is on. For example, dark powders are used on light surface materials and vice versa. Figures 2.14(a) to 2.14(d) illustrate the procedure of collecting fingerprints on a window.

Footprints can also be useful to identify a suspect although they do not have the same significance as fingerprints. Again, the choice of evidence identifiers depends on the material the footprint has been left on. In many cases tape is useful for interiors whereas plaster of Paris or dental stone is useful for soil (see Figs. 2.14(e) et sqq).

Blood traces are categorized as visible and as latent evidence. Because of the specific physical properties of blood it changes color after a certain period of time. It might not be recognized as physical evidence then. The application of chemical enhancement can visualize blood traces. However, this also contributes to the destruction of the blood cells which makes further analysis meaningless. Investigators have to carefully weigh if they should apply chemical enhancement.

Tool marks are caused by tools that modified the surface of a material. For instance, windows or doors may be destroyed during burglary. Collecting evidence then is supported by applying plaster of Paris or dental stone to the impressions. The imprint can then give insight into the applied tool which, in turn, leads to a possible user.

Documentary Evidence

This is provided by written documentation of the crime scene, photographs, sketches, or videotapes. Here, photographic documentation was given the highest priority.

Crime scene photography aims at realistic illustration of the crime scene. The photographer has to capture the crime scene as well as the surroundings under the best circumstances, i.e., by exploiting the lighting conditions. In addition, the photographer has to keep the investigator's view in mind and pictures have to be taken in a certain order. For instance, one sequence would be to take pictures from *periphery-to-center* (in German "von außen nach innen"), taking

- orientation pictures
- overview pictures
- medium pictures
- close-up pictures and pictures of evidence.

Figure 2.15 shows an example sequence. Police forces use single-lens-reflection (SLR) cameras because they feature different photographic formats. Police students are trained to handle SLR cameras. For instance, students have to coordinate different focal lengths, zoom lenses, choose the appropriate film speed, decide whether to use internal, external, or no flashlights, adjust aperture and shutter speed to receive depth of field and sharp images, and so on. Consequently, crime scene photography is crucial when educating police students.

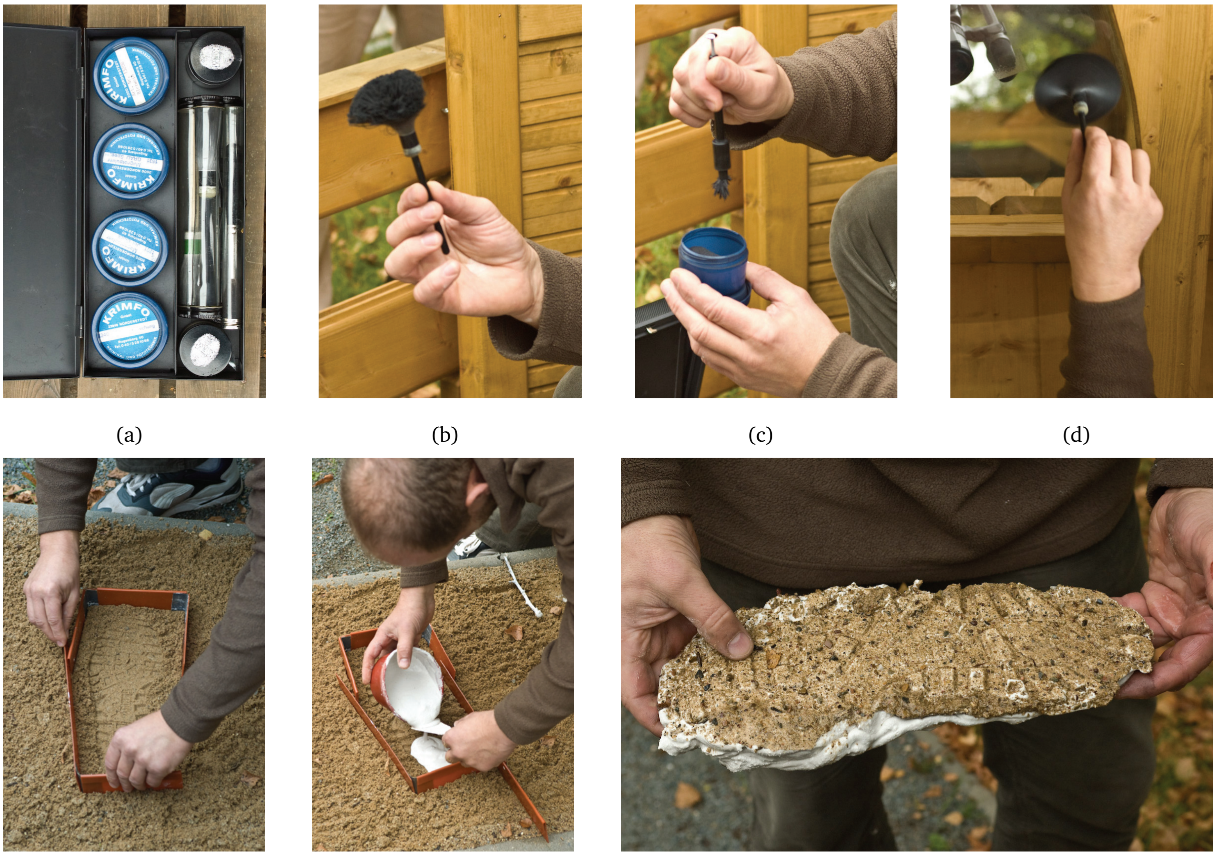


Figure 2.14.: The images illustrate the sequence of measures during the collection of fingerprints (a–d) or footprints (e–g). In the case of fingerprint collection, the investigator decides for the right powder (a) and brush (b) first. This strongly depends on the material the print is on. Then powder and brush are applied (c) and the print is enhanced (d) so that it can be taped and submitted. In the case of footprints, the investigator applies plaster of Paris to the print. Therefore, the print is secured by a form (e), mixture is applied (f), and after some moments the cast can be removed carefully (g). Images courtesy of Jens Grubert.

Having described central aspects of crime scene investigation and CSI education, the next section presents related software applications that concern virtual CSI either as computer game or as training system.

2.4.3 Related Software Applications

This section outlines a couple of related software applications that concern CSI. These generally do not suit the training of police students, though. There is only *SGTAI*, examined closely in Section 2.3.3, that targets training of investigators. However, *SGTAI* specifically concerns accident investigators, whereas the application scenario of this

(a)

(b) (c) (d)

Figure 2.15.: The pictures illustrate four different formats used in crime scene photography: orientation (a), overview (b), medium (c), and close-up image (d). Images courtesy of PHK Peter Eichardt, Police College of Saxony–Anhalt, Germany.

dissertation aims at CSI in general. Any other research projects known to the author of this dissertation so far focus on crime scene reconstruction and use photorealistic computer-aided-design (CAD) techniques [e.g. Gibson and Howard, 2000; Howard et al., 2000; Se and Jasiobedzki, 2005].

Crime Scene Creator is a prototypical software application developed by [Davies et al., 2004]. It is intended to support professional investigators and allows for fast and efficient evaluation of crime scenes. The system can be used to basically reconstruct the crime scene. An animated character can then be added to the scene and can act out the crime according to scene documentation and testimonial evidence. Variable camera views make it possible to view the scenes from different angles. The approach sounds promising, however, no publication documents further development since 2004.

Commercial game titles like, for example, the point-and-click adventure series *Police Quest* or *CSI: Crime Scene Investigation* could lend themselves to encouraging conversations on the game content, but not for interactive training.[17] Also, applications like *Crime*

17 For further information on *Police Quest* see [http://www.vintage-sierra.com/pq.php].
For further information on *CSI: Crime Scene Investigation* see [http://csi.ubi.com/] (March 22, 2009).

Scene Virtual Tour, a Virtual Reality environment for CSI that applies panoramic images of crime scenes, or *Incident Commander*, a training simulation that focuses on crisis management strategies, did not seem to be appropriate training tools for the Police College.[18]

2.5 Summary

This chapter presented a comprehensible insight into the emerging research field of digital game-based learning and digital game-based training systems like serious games. Starting with an historical overview of the field and a brief description of common terminology, the state of the art was presented. The compelling representation of learning content, the attraction to players, and the possibility to structure training sessions have been identified as principal qualities of serious games. The downside of serious games, however, is basically characterized by player-centered game design that replaces teacher and curriculum as well as by a lack of training session reviewing that could support classroom discussion. Recent studies on the use of serious games in education confirmed these difficulties. New software design approaches are required for integrating digital game-based training systems into education. Therefore, educational situations have to be examined. Police education and, in particular, crime scene investigation training was chosen as an application field because it combines practical as well as theoretical educational situations.

The following chapter presents the development of a general framework for digital game-based training systems that faces the challenges posed by recent systems and addresses education as the actual field of application.

18 For further information on *Crime Scene Virtual Tour* see [http://www.crime-scene-vr.com/]. For further information on *Incident Commander* see [http://www.incidentcommander.net/] (March 22, 2009).

Chapter 3

A General Framework for Digital Game-Based Training Systems

In the present chapter a general framework for digital game-based training systems will be developed. This framework provides a solution to support educational settings with digital game-based training systems. Several challenges have to be faced therefore. First, common game design, which primarily supports learning situations, has to be adapted to educational situations (cf. Sec. 3.1). This leads to a schematic representation of game design elements for education and serves as a basis for designing the general framework (cf. Sec. 3.2). A brief summary concludes the chapter (cf. Sec. 3.3). A publication about the general framework can be found in [Brennecke and Schumann, 2009]. It was accepted for GET 2009 and earned positive feedback. All reviewers considered the concept to be novel and relevant for digital game-based learning.

3.1 Adapting Game Design to Education

The last chapter has shown that educational use of serious games is problematic. First, serious games pursue learning situations and thus only address students as game players. Second, serious games hardly provide explicit links to education and hardly provide for adapting game content to current lessons. Third, serious games lack adequate assessment strategies, making judgments on the learning progress of students difficult.

The key question addressed here is, what kind of properties does a digital game-based training system have to consist of in order to suit education? To answer this question, educational settings and game playing situations have to be examined more closely. The educational universe and the game universe are introduced as schematic representations for both situations respectively. A comparison of both universes confirms that game playing, as supported by current serious games, can only depict one part of education. To also depict the other parts of education, game design has to be adapted to education.

3.1.1 The Educational Universe

Education revolves around the acquisition of knowledge and skills through learning, training, and instruction. Here, formal education as implemented by the Anglo-Saxon educational system is the focal point. The key elements identified in the Anglo-Saxony educational system can also be regarded as key elements of informal education in general.

Basically, every educational setting is defined by three distinct sets: teachers $T = \{t_1, \ldots, t_k\}$, students $S = \{s_1, \ldots, s_m\}$, and learning contents $C = \{c_1, \ldots, c_n\}$. In addition, the common educational setting is defined by three relationships: teacher and learning content $R_{TC} = T \times C$, student and learning content $R_{SC} = S \times C$, and teacher and student $R_{TS} = T \times S$. The relationship between teacher and learning content R_{TC} is determined by presentation and instruction, that is, *teaching situations*. The relationship between student and learning content R_{SC} is determined by study and experience, that is, *learning situations*. Finally, the relationship between teacher and student R_{TS} is determined by communication and interaction, that is, *communication situations*. Hence, the educational universe basically describes teaching, learning, and communication. Based thereupon the educational universe EU is defined as schematic representation of education:

$$EU := (T, S, C, R_{TC}, R_{SC}, R_{TS}) \tag{3.1}$$

In pedagogy, the educational universe is well-known as the *didactic triangle*. It is used to explain different forms of didactics [Kansanen and Meri, 1999; Prange, 1983]. Figure 3.1 illustrates the didactic triangle.

It has to be noted that formal education is determined by a curriculum or *specialization of education*. Such a specialization suits the educational level, i.e., grade or institution (vocational school, college, university, ...) and the educational field, i.e., subject matter or specific learning content. This specialization determines *who* will be teacher, *who* will be student, and *what* will be learning content. One part of the specialization is the so-called *syllabus* that structures the learning content into learning units. This is of importance when it comes to training sessions. However, a closer examination of the characteristics of such specializations goes beyond the scope of this dissertation. At this point, it is sufficient to recognize specialization as determining the key elements of education. Yet specialization is not an integral part of the educational universe because it does not characterize education itself.

3.1.2 The Game Universe

Games revolve around contests, challenges, and rewards through which players engage with a conflict. The way players can engage with the conflict is defined by game rules that organize how to proceed with the game [Crawford, 1984; Salen and Zimmerman, 2004]. A common game playing situation or *game setting* therefore is defined by two sets: players $P = \{p_1, \ldots, p_r\}$ and games $G = \{g_1, \ldots, g_s\}$. The game setting is further defined by one relation between players and games $R_{PG} = P \times G$. This relationship is generally determined by experience and problem solving. Similar to the educational universe EU, the game universe GU is also defined as schematic representation of game settings:

$$GU := (P, G, R_{PG}) \tag{3.2}$$

Similarly to the specialization of education, a specialization of game playing exists. It can be considered as game *genre*, determining *what* will be the content field (action, strategy,

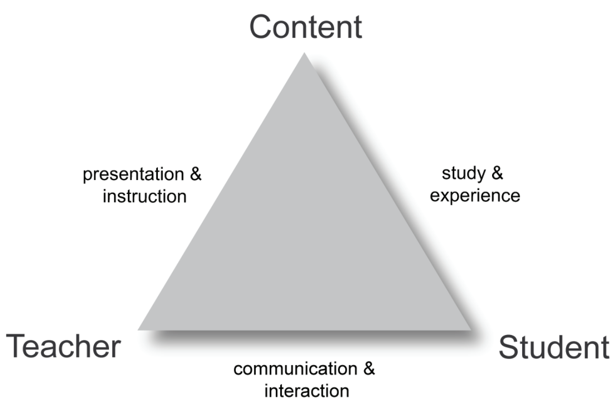


Figure 3.1.: The didactic triangle relates the elements of teacher, student, and learning content on a triangular grid. The relationships depend on the contemporary parameters or *specialization* of formal education.

role playing, ...) and *who* will be addressed. Just as the syllabus structures learning content, game levels structure the game content. However, in contrast to education, most computer games and most serious games pursue a one-player-role design pattern. For this one view of the content and one interaction scheme for content alteration is provided. This strategy depicts only one aspect of education though, as will be explained in the next section.

3.1.3 Relating the Universes

When relating educational universe EU and game universe GU it becomes obvious that EU contains more elements than GU and both universes contain elements such that GU corresponds to a subset of EU. The correspondences are illustrated in Table 3.1.

EU	T	S	C	R_{TC}	R_{SC}	R_{TS}
GU	$\backslash$	P	G	$\backslash$	R_{PG}	$\backslash$

Table 3.1.: Relationships between the educational universe EU and the game universe GU as currently represented by serious games and game-based learning environments.

As was discussed in Section 2.3, game playing is used to reinforce learning situations. The reason for this might be that problem solving, represented by R_{PG}, is closely related to or part of study and experience, represented by R_{SC}. Both relations correspond to each other. As a consequence, players P are used to represent students S. Also, learning content C is represented by game (content) G.

However, learning situations are only one part of education (cf. Sec. 3.1.1). Since game playing relates to learning situations, the game universe omits T, R_{TC}, and R_{TS}. Actually these elements are replaced by the game, which takes care of instruction and content presentation. The extent to which games replace instruction and content presentation certainly is limited and corresponds only vaguely to the significance of T, R_{TC}, and R_{TS} in real world education. In contrast to some authors who tend to promote replacing teachers with games like Shaffer et al. [2005] or Prensky [2003], this dissertation strongly promotes the opposite. Especially in learning situations, understanding and reflection should be strengthened through communication. Therefore, the teacher's role remains essential and has to be included into the educational process. Traditional teaching methods might have to change but should not be replaced.

Therefore, training games have to address T, R_{TC}, and R_{TS} explicitly to support education as a whole. The applications have to let teachers participate in game playing. The applications have to let teachers adapt game content to lessons. And finally, the applications have to support the communication between teacher and student. Therefore, all elements from the educational universe EU have to be mapped to corresponding elements of the game universe GU without losing the educational meaning of T, R_{TC}, and R_{TS}. This requires an extension of GU. As will be explained in the subsequent section, the extension of GU implicitly results in an adaptation of traditional serious game design.

3.1.4 Extending the Game Universe

The extension of the game universe targets finding corresponding elements for T, R_{TC}, and R_{TS}. Set T represents teachers in the educational universe. Teachers are, like students, active participants in education. In the game universe, active participants are represented by game players P. Hence, T corresponds to another set of players P. Since games are not restricted to a specific group of players, set P_T is introduced here as representation of teacher players; set P_S (formerly known as P) is introduced as representation of student players. Along with renaming P to P_S also R_{PG} will be renamed into R_{P_SG}.

Next, relation R_{TC} has to be mapped to a corresponding element in the game universe. R_{TC} represents the relation between teacher and content. The corresponding relation in game universe has to express a relation between teacher player P_T and game G. This is given by $R_{P_TG} = P_T \times G$. Yet what kind of relation does R_{P_TG} represent?

Teachers and students can be regarded as associate opponents, especially in the game context. Both user groups have opposite but associated goals and purposes. These are expressed in R_{TC} and R_{SC}: content presentation and instruction versus content study and experience. Subsequently, R_{P_TG} has to contrast R_{P_SG}. If R_{P_SG} stands for “problem solving”, then R_{P_TG} has to stand for “problem presentation.” In education, problem presentation is commonly part of content presentation and instruction and thus preserves the meaning of R_{TC}.

Finally, relation R_{TS} has to be mapped to the game universe. R_{TS} represents the interaction and communication between teacher and student. In the game universe, this corresponds to interaction between teacher player and student player that is expressed by $R_{P_T P_S} = P_T \times P_S$. Again, what kind of relation does $R_{P_T P_S}$ represent?

In general, communication is essential for human interaction and understanding. In an educational situation, communication and interaction between teacher and student is particularly necessary because teachers usually want to convey information and students want to understand that information. An educational software application should support the communication-interaction-relationship between teacher and student — not only virtually but also in the real world. This could be achieved by reviewing the training sessions and providing a visual feedback as a basis for discussion. The correspondence between the educational universe EU and the extended game universe GU_{ext} is illustrated in Table 3.2.

EU	T	S	C	R_{TC}	R_{SC}	R_{TS}
GU_{ext}	P_T	P_S	G	$R_{P_T G}$	$R_{P_S G}$	$R_{P_T P_S}$

Table 3.2.: Relationships between the educational universe EU and the extended game universe GU_{ext}.

Accordingly, the extended game universe contains the following elements:

$$GU_{ext} := (P_T, P_S, G, R_{P_T G}, R_{P_S G}, R_{P_T P_S}) \tag{3.3}$$

The extended game universe GU_{ext} schematically describes a novel approach of computer game playing for education and, thus, requires an adaptation of conventional (serious) game design patterns: two players have to be addressed and two distinct content interaction paradigms have to be provided. Moreover, the interrelations of the players need to be supported. Here, the resulting software application is considered as *digital game-based training system* because it can be used *to train someone* as well as *to train something* (cf. Sec. 2.2). Yet to implement a digital game-based training system, the schematic representation of GU_{ext} has to be turned into a technical representation first. That is, the design of a general framework is required.

3.2 Design of the General Framework

The design of a general framework turns the schematic representation of the extended game universe GU_{ext} into a technical representation. Such a technical representation or general framework can then serve as a basis for implementing digital game-based training systems. In particular, the general applicability of the framework has to be in focus during the design phase. To ensure a general applicability of the general framework to educational settings, the principal requirement of the framework's design is to incorporate all elements of GU_{ext} such that

- two players P_T and P_S are addressed
- learning content is represented as game G
- relations R_{P_TG}, R_{P_SG}, and $R_{P_TP_S}$ are established.

Central to the technical representation of GU_{ext} are two points: creating the game context G and representing relations $R_i = R_{P_TG}, R_{P_SG}, R_{P_TP_S}$ in it on a technical level. Players P_T and P_S can be neglected, since they are implicitly addressed by the relations R_i.

In addition, the general framework has to be scalable, meaning it has to support specialization (cf. Sec. 3.1.1). It also has to be available, meaning it has to suit different user groups' software knowledge, e.g., experts (probably students) and non-experts (probably teachers). Basically, the technical equivalents of the extended game universe GU_{ext} are illustrated in Table 3.3 and will be discussed in the following sections.

GU_{ext}	P_T	P_S	G	R_{P_TG}	R_{P_SG}	$R_{P_TP_S}$
General framework	Different user types		Data representation	Different interaction paradigms & user interfaces		Reviewing & output data generation

Table 3.3.: The elements of the extended game universe GU_{ext} are related to equivalent elements the general framework has to consist of.

3.2.1 Creating the Game Context

On a technical level, the representation of the extended game universe GU_{ext} has to merge educational context and game context seamlessly (cf. van Eck [2006]). Although this integration is already present in the extended game universe GU_{ext}, it has to be kept in mind that explicit references from game to education are necessary in order not to favor one over the other (cf. Sec. 2.3.5).

Most computer games use *story elements* and *role allocation* to reinforce the player's identification with the game [Blunt, 2006; Gee, 2003]. This is particularly true for adventure and role playing games but also for sports and strategy games where players identify with racing drivers or city mayors, for instance. In terms of digital game-based training systems the learning content has to determine the game story. Since the variety of learning contents is not restricted to some story, no further specifications of the story content can be given at this point. Practical subjects like nursing, fire fighting or crime scene investigation might be transferred to a game story more easily than might theoretical subjects like mathematics, physics, or chemistry. However, games like *Immune Attack* by Kelly et al. [2007] prove that even theoretical subjects can be well transferred to a game story (cf. Sec. 2.3.3).

Provided the game story of some learning content has been determined, it has to comprise *two* distinct game roles to suit players P_T and P_S. The goals and ambitions of the game roles have to conform to authoring, i.e., problem presentation R_{P_TG}, and training, i.e., problem solving R_{P_SG}, accordingly. Hence, goals and ambitions of the player roles have to oppose each other (cf. Sec. 3.1.4).

Generally, game stories contain some kind of *conflict* that challenges the player throughout the game story creation [Crawford, 1984; Salen and Zimmerman, 2004]. In case of a digital game-based training system the conflict has to be split into *two* conflicts because *two* roles with opposing goals and ambitions exist. The two conflicts represent relations R_{P_TG} and R_{P_SG} in the context of the game story.

Reconsidering relations R_{P_TG} and R_{P_SG} in the context of education, R_{P_TG} corresponded to *teaching situations* and *problem presentation* whereas R_{P_SG} corresponded to *learning situations* and *problem solving* (cf. Sec. 3.1.1 and Sec. 3.1.4). Both descriptions denote equal situations, regarding the prevailing context. On the account of game stories the two relations lead to two different parts of the story: *back-story* and *main-story*. The back-story is a common feature in computer games that introduces the game player to the starting point of the game (cf. Blunt [2006, p. 58] or Ryan [1999, cf. "Storytelling"]). Here, the idea is to make the back-story an independent part of the game by which the teacher player P_T can create the training session. The result of the back-story then can be used as a basis and as beginning of the main-story. Hence, the traditional game story has to be extended to also containing its pre-plot. At the same time, however, the back-story must contain an independent conflict to challenge the player and to conform to a game story [Ryan, 1999].

The main-story, on the other hand, resembles what has traditionally been the story part of serious games. It has to be played by the student player P_S. Like the back-story, the main-

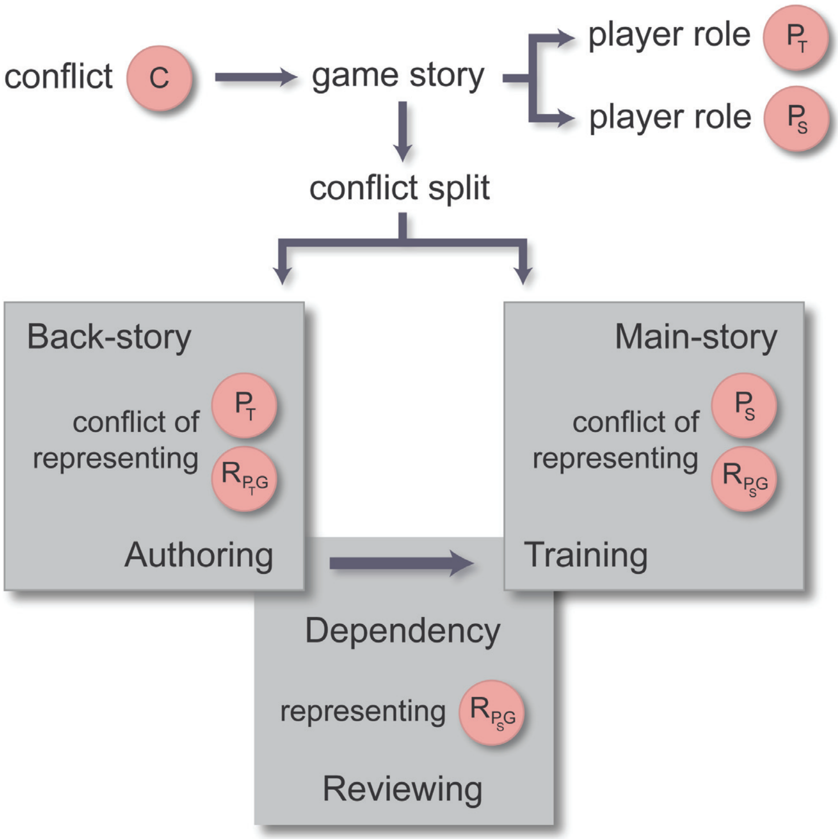


Figure 3.2.: To create the game context, the game story conflict has to be split into two conflicts and two game players have to be addressed, accordingly. This leads to two sub-stories: the back-story specifies the main-story. It represents teaching situations and, thus, corresponds to *authoring* game content. In contrast, the main-story represents learning situations and corresponds to *training*. Both stories depend on each other and allow for fostering communication & interaction situations by *reviewing* the game sessions.

story has to contain an own conflict which is *specified* by the prevailing back-story. Hence, when transferring back-story and main-story into a technical context, both stories imply and require two different interaction paradigms. Relation R_{P_TG} expresses *authoring* of training sessions whereas relation R_{P_SG} expresses *training*, i.e., experiencing training sessions.

The dependency, which connects back-story and main-story, expresses the relation between the player roles of P_T and P_S and implies relation $R_{P_TP_S}$. This relation $R_{P_TP_S}$ could further be expressed by reviewing game levels, i.e., training sessions. An automatic and visual review could assess the training session and serve for the teacher's evaluation or self-evaluation as well as a basis for discussion in the context of education. The creation of the game context is illustrated in Figure 3.2. The resulting framework overview will be introduced and discussed subsequently.

3.2.2 Framework Overview

The creation of the game context presented in the previous section leads to the general framework that will be introduced here. In order to transfer the concepts of back-story (authoring), main-story (training), and dependency (reviewing) to a general framework, three components are necessary: *authoring component*, *training component*, and *reviewing component*. Figure 3.3 illustrates the general framework for digital game-based training systems. The presented structure of the general framework represents all elements of the extended game universe GU_{ext}. The principal form of the framework will be briefly examined below and discussed in greater detail in the following chapter.

Data Presentation

The presentation of learning content basically is a presentation of data in a game-based user interface. The general framework proposes integrating existing learning material to smoothly link real world education and digital game-based training system.

Internal Data Such data comprise general *game story data* like scenery and environment as well as *player-specific data* like player character or specific user interfaces. Game story data serve as the basis for back-story and main-story; they have to be accessible by the authoring as well as by the training components. Player-specific data have to be encapsulated in the authoring or training component. Especially, the user interfaces have to support the respective game story in terms of authoring and training. Game levels are used to encapsulate game story data and to organize the training sessions, e.g., in accordance with the syllabus (cf. Sec. 3.1.1). As an alternative, player-specific data could also be stored in the game levels. This would result in a highly generic system structure. However, seen from the implementation point of view, the separation of player-specific data and respective component would be difficult to implement. This is due to the fact that functionality provided by the components to handle player-specific data would have to generically adapt to different player-specific data.

External Data Principally, these data denote existing learning material like texts, images, or video data. Such data could be used by teachers to enhance the back-story and pose problems. They could be used by students to accomplish tasks and solve problems. External data also have to be stored in the game levels.

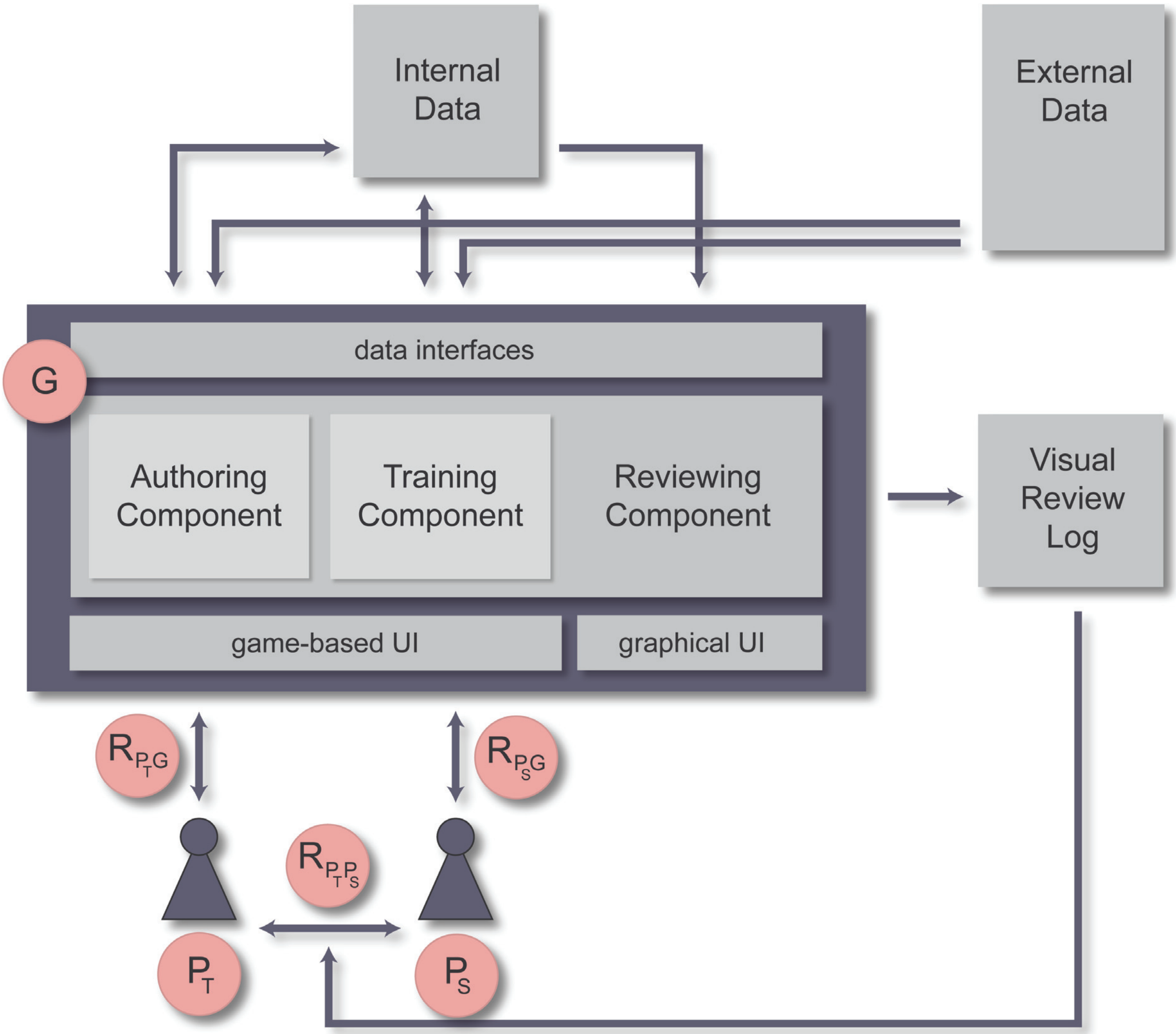


Figure 3.3.: The general framework for digital game-based training system is divided into three main components for authoring, training, and reviewing. Data interfaces allow for data integration and access. Game-based user interfaces allow the game players to interact with the respective game content and to play the game stories. To specify the reviewing process, another graphical user interface is provided. The visual review log can serve as a basis for classroom discussion, for example. The general framework is based on the extended game universe and its elements, as depicted by the red circles.

Authoring Component

The authoring component addresses teacher player P_T and represents relation R_{P_TG} in a game context. It has to provide interaction means for *authoring* training sessions by playing the back-story. This kind of authoring through game playing will be introduced here as *in-game authoring*.

In-game Authoring In-game authoring defines the interaction paradigm of authoring as a game playing experience. In-game authoring has to acquaint non-experts with authoring training sessions by specifying the main story of a two-part story-based digital game-based training system.

In-game authoring primarily requires an appropriate user interface design that supports

- playing the back-story
- accessing and integrating external data
- creating tasks.

A game-based user interface for playing the back-story should facilitate the application use, as was outlined in Section 2.3.1. It has to be determined in accordance with the back-story. Apart from the game-based user interface, the authoring component has to provide two further interface types: task interfaces and interfaces for external data integration. A combination of both might be useful. External material could then be integrated to support task accomplishment. The possibility of inserting training aids would also support scalability of the digital game-based training system in terms of specialization because the teacher could determine how much help is granted and how much self-effort has to be afforded when using the training component.

Training Component

In contrast to the authoring component, the training addresses the student player P_S and represents relation $R_{P_S G}$. The component conforms to traditional serious games. Like for the authoring component, the interaction paradigm of *in-game training* will be introduced here.

In-game Training In-game training defines the interaction paradigm of training as a game playing experience. In-game training has to acquaint expert game players with using a digital game-based training system as educational tool. Therefore, in-game training has to be supported with explicit links of game story and real world education.

Like in-game authoring, in-game training requires an appropriate user interface design that supports

- playing the main-story

- accessing and integrating external data
- accomplishing tasks.

Basically, the training component has to mirror the authoring component. It has to allow for playing the main-story on the basis of the back-story. Therefore, a specific game-based user interface is required. Moreover, the training component has to allow for integrating external material if required by the training session or task. In this context, it also has to allow for an explicit interface for accomplishing created tasks. Most importantly, the training component has to contain game elements that motivate the students to use it. This will be discussed in greater detail in Section 4.4.

On the basis of authoring and training, the reviewing component has to assess the training session. This will be examined next.

Reviewing Component

The reviewing component of the general framework addresses players P_T and P_S and represents relation $R_{P_T P_S}$. Since this is a human relation, the technical representation is not straightforward. Regarding relations $R_{P_T G}$ and $R_{P_S G}$, these were technically established by designing an interface that supports player-game-interaction. In that regard, the technical establishment of $R_{P_T P_S}$ has to provide "something" that supports player-player-interaction and communication. The assessment of game playing, i.e., of the training session and the generation of visual review log was proposed for this. The reviewing component thus has to implement

- logging of specified data and data changes such as those caused by user interaction
- evaluation of the logged data in terms of training session assessment
- visualization of the assessment in a comprehensible visual review log.

The generation of a visual review log is not trivial because it requires specification, logging, and assessment of relevant data before the visualization can be conducted. Here, user interaction with the scene is regarded as relevant data to start with. At best, user interaction that is connected to task fulfillment has to be logged. Based on the logged data, a visualization is required. Therefore, snapshots of the scene as well as additional visual clues could be used. Scene snapshots could then be arranged in a chronological order, for example, mimicking storyboards or comic strips. To convey further information, additional visual clues like storyboard symbols [Goldman et al., 2006] or non-photorealistic renditions of scene objects [Strothotte and Schlechtweg, 2002] could be added to highlight certain user interactions with the scene. To allow users to specify which data to log and how to visualize these data, a graphical user interface will be required here.

All in all, reviewing training sessions is complex and demands considerable and novel developments, as will be discussed more profoundly in Section 4.5.

3.3 Summary

The current chapter presented the development of a general framework for digital game-based training systems. This framework was carefully designed to suit educational situations. Schematic representations of common educational situations as well as of common game playing situations were introduced to illustrate their differences — especially, in regard to current serious games' design. The comparison resulted in an extension of the game universe to better suit education by addressing learning situations as well as teaching and communication situations. In Section 3.2, the schematic representation of the extended game universe was then transferred to a technical representation, i.e., the general framework for digital game-based training systems. The general framework converts teaching, learning, and communication into the technical equivalents of authoring, training, and reviewing. This way, the general framework extends common serious game design and presents a novel approach for developing digital game-based training systems by addressing the educational situation. As a consequence, the general framework raises several research questions, which the subsequent chapter will go into. Especially, the difficulties and challenges of turning the general framework into a concrete software application will be discussed there.

Chapter 4

Design Issues — From Framework to Implementation

If the last chapter discussed *what* necessities a digital game-based training system requires, this chapter discusses solutions for *how* to build a digital game-based training system. More precisely, the chapter examines a wide range of challenges that have to faced during the design phase. Starting with general considerations (cf. Sec. 4.1), the chapter draws on technical challenges like data representation (cf. Sec. 4.2), authoring, training, and reviewing components (cf. Secs. 4.3, 4.4, and 4.5). Finally, the chapter examines how to balance learning and gaming to be most effective and presents a workflow schedule that describes necessary steps to implement a digital game-based training system (cf. Secs. 4.6 and 4.7).

4.1 General Considerations

When designing a digital game-based training system the designer or developer has to keep in mind that the digital game-based training system is a supplement to education. The premier goal of the system therefore has to be supporting the user in handling the learning content. Hence, user-centered design is of particular interest for digital game-based training systems. To achieve this, different design strategies can be pursued [Saffer, 2006, Chap. 2]. Moreover, instructional system design methods are generally applied when developing educational software. In particular, ADDIE model is well-known as a development guideline [Quinn, 2005]. The model is divided into five phases: analysis, design, development, implementation, and evaluation. As Quinn [2005, p. 123] points out, “the development phase creates the materials, the implementation deploys them, and evaluation determines their effectiveness.”

ADDIE model is a good starting point to structure development processes. However, the model adds no further details to the phases and remains abstract. In contrast, this chapter aims at establishing a workflow schedule that is formed to the requirements of developing digital game-based training systems. The workflow schedule targets supplementing the general framework by indicating difficulties and demands on the implementation. It is intended to support developers of digital game-based training systems and will contain specific details.

As was already mentioned, user-centered design is of particular interest for digital game-based training systems. Each interaction component of the general framework addresses distinct user groups: teacher, student, and both. Each user group has to work with the same learning content or data, but each user group has different working demands. Teachers pursue teaching situations whereas students ideally pursue learning and training. Thus, each user group needs distinct views of the data and distinct techniques for data handling. The following subsections therefore discuss requirements of the user interface regarding data presentation and data handling.

4.2 Data Presentation

The learning content of a digital game-based training system has to be reflected in the game story. In turn, the game story determines the user interface as well as necessary game story data. To substantiate the seriousness of a digital game-based training system and, thus, to substantiate the link between game story and real world education, the data component has to present game story data authentically. Two questions result from there:

- How to emphasize the data’s authenticity?
- How to emphasize the digital game-based training system’s value for education?

The struggle for authenticity not only concerns the data themselves. Also, the application has to be considered an authentic educational tool (cf. Sec. 2.3.5). Based on McLuhan [2001], another question might also add to the discussion: Regarding a digital game-based training system as an interactive medium, how can it transfer an educational message?

Answering these questions greatly depends on the specific learning content, resulting game story, and digital game-based training system. The developer will have to discuss the questions on the basis of the given subject matter. Yet aspects of authentically presenting data can well be discussed at this point. In this dissertation, central authenticity supporters are identified as

- data representation
- data depiction
- data behavior in response to user input.

Each aspect will be discussed below. Data representation as well as data depiction have already been thoroughly examined in research whereas data behavior has rarely been discussed as authenticity enhancer for digital game-based learning.

4.2.1 Data Representation

Data representation depends on the technology used and can almost always be divided into 2D and 3D representation. 3D representation of data is often considered a real-world-metaphor because it allows for realistically constructing 3D geometry and virtual worlds [Teyseyre and Campo, 2009]. However, depending on the specific learning content, a 2D representation of the data might be preferable. For example, 2D representations could be used to abstract from realistic 3D geometry and focus on certain aspects only. It could be used for transferring an educational message. Research on user interface design provides insightful discussions on the different representations [Ark et al., 1998; Baumgärtner et al., 2007; Cockburn, 2004; Cockburn and McKenzie, 2002; Tavanti and MatsLind, 2001].

Ware [2001] discusses human perception in respect of 2D and 3D representation. According to the author "...the 3D information provided should depend on the application and the task" [Ware, 2001, p. 255]. The author suggests a hybrid approach he calls *2 ½D representations*. That is, for example, using visible 2D navigation cues to support user navigation in 3D virtual environments. In digital game-based learning scenarios, such combinations of data representation could be used to enhance realistic data representation with additional educational messages. For instance, some meta-information could be introduced that carry the educational messages. In this context, Teyseyre and Campo [2009] are recommended for further reading. The authors thoroughly discuss 3D data representation for 3D visualization.

Generally, developer and designer have to determine which representation supports learning content and game story best. Since most current (serious) computer games predominantly use 3D representations of the content, this dissertation primarily puts emphasis on 3D representations.[1]

4.2.2 Data Depiction

If data representation is concerned with how to describe the data technically, data depiction is concerned with how to visualize these descriptions. Authentic data depiction is often connected with computer graphics techniques that enhance photorealistic rendering of 3D scenes. Photorealism aims at *objectively* representing the external world and, thus, obviously links virtual world with reality [Fun et al., 2004]. However, perfectly rendered photorealistic images are still expensive and often times can not be used in real-time; imperfect photorealistic images distract users and might cast doubt on the authenticity (cf. Egenfeldt-Nielsen [2005]). Additionally, photorealism rarely provides visual means to convey additional information like an educational message.

In contrast, non-photorealistic rendering (NPR) imitates artistic expressions like line drawings or hatching to depict and illustrate data [Gooch and Gooch, 2001; Strothotte and Schlechtweg, 2002]. NPR techniques can be used to emphasize or omit detail and, thus, steer the viewer's attention [Viola and Sousa, 2006]. Hence, non-photorealism is often applied for illustrative purposes, such as to effectively communicate information visually [Ebert et al., 2005; Sousa, 2003; Strothotte and Schlechtweg, 2002]. Since focusing user attention is an essential part of education, non-photorealistic rendering techniques seem to be fairly adequate for digital game-based training systems.

In terms of authentic data depiction the question arises, if non-photorealistic depiction decreases authenticity. This should be investigated further. Whether to deploy photorealistic or non-photorealistic rendering will most certainly also depend on task, data,

1 To be more precise, most serious game applications employ 2D as well as 3D representations. 2D representations are commonly used for the graphical user interfaces like buttons, etc. Yet 3D representations are commonly used for representing the game content itself which is why they are referred to as using 3D representations.

and educational message to be transferred. Photorealistic rendering might serve well for visualizing facts. Non-photorealistic rendering might serve well for focusing user attention and for stimulating reflection and understanding. For instance, a combination of both rendering styles seems to be predestined to depict data information and data's meta-information. At least, the rendering style should be carefully specified in order to support the game story.

4.2.3 Data Behavior

Data behavior has been identified as the third aspect of authentic data presentation. Authentic data behavior is given if data modifications caused by user input result in realistic changes of the data. This requires the data representation, first, to contain authentic data properties and, second, to adapt the properties realistically when modified. Moreover, interactive and real-time response to user input is regarded as a crucial enhancement factor for supporting situated learning [Schneider and Godard, 1996]. As with unrealistic data depiction, unrealistic data behavior will support the user's doubt of the applications support to education.

Data behavior tends to be a stronger indicator of authenticity than data depiction though. As observed by BinSubaih et al. [2008], data themselves do not necessarily have to be as in real life. Yet situations that are represented by data have to be as in real life to achieve the effect of authenticity: BinSubaih et al. report that a menu-based dialog of *SGTAI* was considered to be a reality improvement because it reproduced an authentic dialog situation [BinSubaih et al., 2008, p. 28]. The observation shows that the effect achieved by how the data "behave" and how the data are employed can make up for unrealistic data, e.g., using a menu-based dialog for reproducing conversation (cf. also Harteveld and Bidarra [2007]). Therefore, authentically reproducing how data behave should be prior to authentically depicting how data look. This should also be of kept in mind when regarding the given resources for developing a digital game-based training system. Nevertheless, it would be of interest to investigate whether a virtual dialog between two game player characters would further support the impression of authenticity.

4.3 Authoring Component

In general, authoring is a non-trivial interaction paradigm. This is particularly true for *in-game authoring*, which was introduced in this dissertation as promoting authoring-by-game-playing. Still, the authoring component has to establish in-game authoring in such a way that it supports teaching situations. Two questions arise from this first of all:

- How can in-game authoring support teachers and teaching situations?
- How can in-game authoring acquaint teachers with computer games playing?

Although literature has thoroughly discussed learning-by-game-playing, authoring-by-game-playing is a novel approach. A few existing authoring approaches seem relevant to developing in-game authoring. These will be discussed below.

4.3.1 Relevant Approaches

Authoring systems are applications used for creating other applications like virtual environments, games or game levels. Authoring often becomes highly complex, e.g., when modeling 3D geometry, arranging a virtual scene, adding behavior to objects, etc. is necessary. This is especially difficult for non-experts [Good et al., 2006; Nelson and Mateas, 2008; Ryan, 1999]. Consequently, overall design and creation of a training system should be crafted by professionals (cf. also van Eck [2006]). To acquaint non-experts with authoring, they should be given authoring methods that allow them to specify existing training or game settings. *Modding* seems to be relevant here as do methods from real life role playing games.

Modding denotes authoring of computer game levels and new games by using game editors that come with commercial computer games [Behr, 2008]. For example, *World of Warcraft III* ships with an editor to author new game levels. The author can select game elements, arrange them in the game world, and add behavior as well as properties to the elements. In digital game-based learning, modding was successfully applied to support learning by game building [El-Nasr and Smith, 2006; Korte et al., 2007; Robertson and Good, 2005; Szilas et al., 2007].

Modding often times presupposes advanced knowledge in computer game playing. This seems to be the strength of modding. Players who are familiar with the game world, its elements, user interfaces and interaction means are easily accustomed to authoring this world. Hence, modding seems to be a good starting point for determining properties of in-game authoring.

On the other hand, real life role playing games provide techniques for mastering the game flow. Flowers et al. [2006] examined different strategies of real life role playing game masters to direct the story flow. Story flow direction can be considered a specification technique relevant to a digital game-based training system. For instance, game masters commonly use threats versus rewards to lead players away from or to certain situations. Such game mastering strategies seem suitable to support students in accomplishing tasks or in finding appropriate context information. Moreover, they can be considered in-game strategies and, thus, fit well with in-game authoring.

Modding seems to be a suitable approach for designing the authoring component for two reasons: first, modding is tightly connected with the underlying game and allows for smoothly accustoming users. Second, modding is regarded as a creative process for non-expert authors to adapt and to further develop computer games [Behr, 2008]. Support for adapting an existing training system to a specific lesson was necessary to support teachers (cf. Chap. 3) and makes modding features desirable for digital game-based training systems. An additional integration of game mastering techniques would further strengthen contextual adaptation of the training session to a specific lessons. Modding can be regarded as supporting interaction with and authoring of the game world, whereas game mastering can be regarded as supporting authoring of the training strategies. Both approaches should be integrated into in-game authoring as discussed next.

4.3.2 In-Game Authoring

In-game authoring has been introduced as an authoring paradigm to produce a specific training session by playing and altering the back-story. The following two authoring approaches will be introduced in this dissertation to integrate modding and game mastering techniques to digital game-based training systems:

- explicit authoring
- implicit authoring.

The distinction between explicit and implicit authoring shall not be considered obsolete but rather as an indication. A hybrid authoring approach that contains explicit and implicit techniques will probably most effective.

Explicit Authoring Explicit authoring resembles traditional authoring techniques: the user is given a set of options for game editing. Here, editing options primarily concern the integration of external material and task creation. Also, explicit authoring could additionally concern modification of scene objects as well as removal or addition of scene objects. The respective user interfaces depend on the type of material, e.g., text, images, or video, as well as on the type of tasks, e.g., multiple choice tests, instructions about how to proceed with the game, or instructions how to interact with the scene. User interface design will result in several questions the developer has to answer like,

- How to integrate external text/images/video into the game story?
- Does an integration increase or decrease the system's authenticity?
- What kind of task representation will be necessary to support teaching?
- Should the authoring editor be integrated into the actual game world or would an explicit authoring environment be preferable?

Still, these questions can only be answered in the respective learning and game context.

Implicit Authoring Implicit authoring regards the authoring process as a game playing process. Hence, the teacher should not be aware of authoring but rather of playing the back-story. This relates to the game mastering techniques because game masters also participate in game playing. To implement implicit authoring, the back-story has to be carefully designed such that it

- tightly connects to the main-story
- guides the user in accomplishing task setups
- uses game mastering elements to direct the user.

The back-story furthermore will require an underlying task representation, which generates tasks automatically in correspondence to the user's interactions. Several further questions arise from there:

- How to set up tasks implicitly through game playing?

- How to link back-story and main-story?
- How to apply game mastering elements to the back-story without restricting the user's freedom of explicit authoring?

Answering these questions might require consulting a story developer.

Combining explicit and implicit authoring seems to be a desirable approach for in-game authoring. For example, modding as well as game mastering techniques could be made part of explicit authoring as well as of implicit authoring. In the first case, an explicit editing interface would be required. In the second case, the user would have to edit the game through playing and, thus, accomplish implicit authoring. For explicit authoring, the author should always be aware of what is authored whereas for implicit authoring, the author should be aware of playing the back-story. Therefore, implicit authoring requires the underlying system to be designed in such a way that it allows for game world as well as for game story modification.

4.4 Training Component

The training component addresses the user group of students and has to support training situations. Since most students are referred to as so-called "digital natives," they can be assumed to be experienced software application users [Gee, 2003; Prensky, 2001; van Eck, 2006]. The main challenge of the training component will, thus, not be smoothly acquainting students with computer game playing. Rather, challenges will be on motivating students to continue playing and help students to recognize the digital game-based training system as an educational tool, as was already discussed in Section 2.3.5.

Relevant approaches for implementing training games have been discussed in-depth in Chapter 2. Hence, technical aspects of how to motivate students will be in focus next, before examining technical aspects of in-game training.

4.4.1 Motivation for Training

Motivational aspects of game playing are manifold. They comprise game story, role identification, game-based interfaces, combination of challenge and reward, etc. Central to education is self-motivation of students. According to Ryan and Deci [2000], three aspects are considered central in supporting human self-motivation:[2]

- competence
- autonomy
- relatedness.

According to the authors, feelings of competence caused by (positive) feedback, communication, and rewards can increase and foster motivation, if these feelings are accompanied by feelings of autonomy. Humans need to experience self-determination in what they are doing. Additionally, a secure and familiar setting is said to provide further motivation.

All three aspects can be mapped to game playing and story design of the main story. Competence could easily be conduced by carefully providing for positive feedbacks and rewards during the course of training. Communication could furthermore be increased, if the characteristics of the game roles of back-story and main-story were strongly related. For example, if teacher and student played opponent roles or accomplice roles, communicative elements caused by competition or teamwork could be integrated into the game.

Autonomy is generally an integral part of computer games. Still the system has to provide means that let players perceive themselves as self-determined when playing the game (cf. Paras and Bizzocchi [2005]). This could be fostered by comprehensible and elaborate game-based interfaces with which the player can modify and proceed with the game.

Finally, relatedness should particularly be made present in digital game-based training systems using two facts:

1. The student player has to relate to the underlying learning content that is represented by the game story.
2. The student's player role has to relate to the opponent or accomplice that is determined by the teacher's player role.

2 In psychology, motivation is divided into *intrinsic motivation* and *extrinsic motivation*. Intrinsic motivation is considered to be "... the inherent tendency (of human nature) to seek out novelty and challenges, ..., to explore, and to learn" [Ryan and Deci, 2000, p. 70] whereas extrinsic motivation is considered to be referring "... to the performance of an activity in order to attain some separable outcome" [Ryan and Deci, 2000, p. 71]. Both aspects are considered central factors that motivate humans to pursue their goals.

The role characteristics could also be used to influence game mastering strategies like threats or rewards (cf. Sec. 4.3). The motivational elements primarily concern the development of the game story. Yet they should also be kept in mind when designing the training interfaces for in-game training.

4.4.2 In-Game Training

In-game training has been introduced as a training paradigm that is based on playing an educational game story. Hence, training is supported using game-based elements like a digital game-based training system. A basic requirement for in-game training has been

- emphasizing explicit links to real world education.

Such explicit links have to support students in recognizing the training game as an educational supplement rather than as an entertaining game only. Such explicit links could be

- linking game tasks to external material, e.g., textbooks by
  - consulting external material to advance a game level
  - integrating external material to solve a game task
  - exporting game data for group work, presentation, or classroom discussion.

For instance, students could have to prepare some group work or presentation after having played some game level. If the game level contents covered distinct aspects of the learning content, several presentations could be prepared. Also, students could recapitulate other student's presentations by playing the respective game level. To motivate students to prepare good presentations, high scores could further be used as an explicit motivational element. Again, several design questions will arise, for example,

- How to interlink external material and game task?
- How to support competence and autonomy when integrating external material?
- How to support student's group work using the digital game-based training system?

Apart from supporting explicit links between training and gaming, the training component has to integrate training strategies into the main-story. Story-telling professionals could be collaborated with for further support. The main-story should be designed such that it

- integrates training strategies into game playing
- is tightly based on the back-story in order to
  - support motivation to play with or against the teacher
  - provide for a coherent presentation of implicitly authored tasks.

Training should support the relationship between teacher and student and strengthen their communication within the setting of the game stories. Moreover, the training component has to integrate external material into the main-story's structure to provide explicit links to education.

Finally, the reviewing component is intended to substantiate communication with assessment facts.

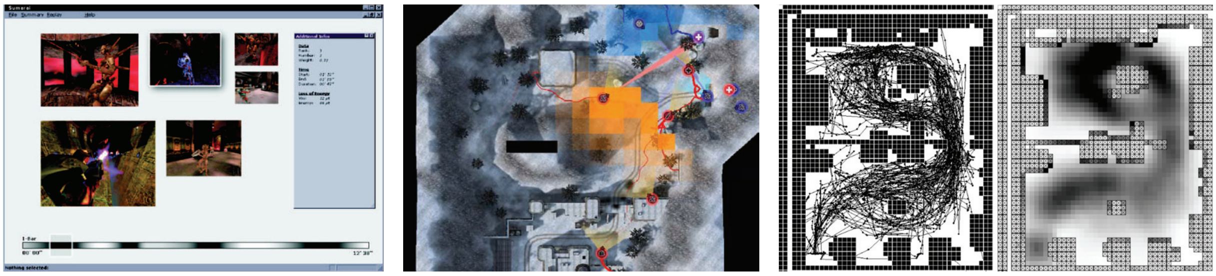
4.5 Reviewing Component

The reviewing component addresses both user groups, teachers and students, and has to support communication situations. As a basis for communication, assessment of training sessions in a visual form was already proposed (cf. Sec. 3.2.2). The automatic generation of training session assessments and an additional visualization of the assessments comprises a range of different research topics. The present section will first discuss related works on this topic before focusing on the design issues that can be deduced from this.

4.5.1 Related Work

Replaying functionality is a common feature in most computer and serious games. It allows for reproducing game sessions as video sequences. However, as an evaluation of user interaction is omitted in most cases, replays do not seem to be suitable as training session summaries [Sickel, 2006]. In contrast, academic approaches have targeted logging and evaluating user interactions. The visualizations, however, lack readability at least for non-experts because they seldom include common visual cues from art or narratives.

Halper and Masuch [2003] present so-called *action summaries* of action games. The authors analyze game state variables over time and visualize them as a series of game



(a) [Halper and Masuch, 2003] (b) [Hoobler et al., 2004] (c) [Chittaro and Ieronutti, 2004]

Figure 4.1.: Summaries of user interaction in virtual environments. Halper and Masuch [2003] sequentially summarize snapshots of the game that have been identified as representative game shots (a). Hoobler et al. [2004] depict current situations of the game, e.g., the image illustrates where the two opponent groups (orange and blue) are positioned (b). Chittaro and Ieronutti [2004] visualize regions of interest illustrated as dark area (right) by evaluating user navigations paths (left) (c).

snapshots (see Fig. 4.1(a)). Game state variables could be used together with tasks to evaluate task accomplishment. However, the action summaries approach is restricted to taking snapshots only. Highlighting specific events is disregarded as is the establishment of contextual relationships between snapshots. In contrast, Chittaro and Ieronutti [2004] and Hoobler et al. [2004] visualize overall user activities highlighting user navigation paths (see Figs. 4.1(b) and 4.1(c)). Hoobler et al. [2004] additionally encode action over time into colored game area overlays to illustrate current game situations. Yet this takes into account topical activities only. Grammenos et al. [2006] have also studied user activity in virtual environments. They introduce the *virtual prints* as a concept to trace user navigation and interaction. Additionally, users can leave marks in the virtual environment to communicate with each other. Such visualization could be used to illustrate the course of the training session. Moreover, visual communication marks fit the dependency between back-story and main-story. In the case of accomplice game roles, the teacher could leave hints in the virtual environment (cf. Sec. 4.4.1).

Although generating replay-like *movie summaries*, Friedman et al. [2004] put emphasis on story understanding by chronologically relating specific events from virtual environments. To do so, objects and actions of interest are manually specified and automatically tagged by a filtering algorithm. The movie summary is then generated under consideration of time constraints. Fielding et al. [2006] also target the summary of specific events from virtual environments on the basis of user interest. The authors introduce reporter, editor and presenter agents. Each of them is intended to log, filter and visualize important events according to specified user interest. Implementation details or examples are undocumented, however.

(a) [Goldman et al., 2006] (b) [Nienhaus and Döllner, 2005]

Figure 4.2.: The illustrations are of interest as additional visual cues in review logs. Goldman et al. [2006] apply style elements from storyboarding to visualize the sequence of events, i.e., to summarize the woman's running path (a). Nienhaus and Döllner [2005] use visual cues from narrative arts and storyboards to depict dynamics in images of 3D scenes (b). Here, three different visualization styles depict a flying ball.

Apart from reviewing user interaction in virtual environments and games, user interaction logs are also of relevance in different research domains. These will briefly be outlined below because they seem equally suitable for designing the reviewing component. Tominski [2006] aims at enhancing information visualization by highlighting data changes, so-called *events*, of user interest. Users can specify certain event types which are detected and represented by the visualization. In digital game-based training systems, such an event could, for example, be a user interaction that changes state from "button pressed" to "button released" of some object.

Alongside data specification, data presentation is of equal importance for generating a review log. Goldman et al. [2006] present an approach for schematic storyboarding for video visualization and editing (see Fig. 4.2(a)). The authors discuss the schematic language of storyboards and apply it to manually selected video sequences. This way, the authors allow for summarizing video films by preserving the course of the story. In contrast to storyboarding, Gordon [2006] and Kurlander et al. [1996] use comic strip formats for visualizing communication between virtual characters. Nienhaus and Döllner [2005] developed an approach to depict dynamics in 3D animations that is based on visual art and narrative comics as well as storyboards (see Fig. 4.2(b)). The authors analyze the 3D scene and its dynamics using 3D scene graphs. Furthermore, they introduce the concept of dynamic glyphs, i.e., additional graphics elements used to depict dynamics in images of 3D scenes. The approaches seem to be of high relevance for visualizing the course, chronology, and dynamics of the training sessions.

4.5.2 Reviewing Training Sessions

The previous section showed the breadth of relevant topics that have to be examined to deliberately design a reviewing component. Such an elaborate concept surpasses this dissertation's scope. Instead, central design challenges will be discussed below that serve as a basis for future developments. Such challenges comprise

- specification and logging of relevant data
- evaluation of data for assessment purposes
- visualization of the assessed data as a visual review log.

Data Specification

The main goal of a reviewing component is to present to the user a comprehensible summary of the training session that highlights important aspects of training. Yet to review training sessions, data have to be specified the underlying reviewing component can assess. In this context, it is helpful to regard a digital game-based training system as an event driven state machine, which is a regular notion of computer games (cf. Jool [2005]) and holds true for digital game-based training systems.[3] This means, each training session is considered to be in a certain system state, i.e., a specific scenario setup that has a specific parameterization of scene objects, etc. According to Tominski [2006],

> "...an event is considered, if conditions that users are interested in are suddenly all met in the data. In this sense, events can be considered special portions of data that are of particular interest, because they match with some interest expressed by users." [Tominski, 2006, p. 44]

Hence, by interacting with the scene, the user can change parameter settings and trigger an event. On the one hand, this would result in a new system state. On the other hand, this could be used to trace back the user's behavior during training. Tominski [2006] defines different types of events like simple events or composited events. Event data could be a simple text message that is sent through the system like "button pressed" or a complex concatenation of different events that could, for example, be used to summarize

3 For further information on finite state machines and event driven state machines see Cassandras and Lafortune [1999] or Wagner et al. [2006].

a sequence of user interactions. Thus, event data could serve as a basis to trace back user interactions.

Two further aspects have to be considered in the context of data specification. First, data that is relevant to assess the training session will be referred to as *assessment data*. Yet since the reviewing component should additionally produce a visual illustration of training session assessment, further data have to be specified that are relevant for training session visualization. Such data will be referred to as *visualization data*.

Assessment Data Task accomplishment is of primary interest for training session reviewing. Therefore, a task representation that uses certain task events to assess task accomplishment or failure should be specified. Furthermore, to trace back general user behavior during training, specific user interaction events might also be specified. For example, if the user's exploration of the virtual environment is of interest, certain user interaction events could be triggered by user navigation. Generally, the data required for assessing the training session depend on the specific field of education and can hardly be specified here. Moreover, a virtual 3D environment might call for different assessment data than does a 2D application for processing quizzes. Here, relevant assessment data are determined by

- task events that describe
    - accomplished tasks
    - abandoned tasks
    - disregarded tasks
- special user interactions events that describe
    - user navigation
    - scene alteration.

Visual Data In addition to the assessment data, data are required that help illustrate the training session and illustrate the user interactions. Like assessment data, visual data depend on the technology used. Commonly, *time* is relevant for tracing back assessment data chronologically. Other visuals have to be found that determine *how* to emphasize assessment data.

In the case of a 3D interactive environment, *player positions* are relevant to trace back the user's navigation path. Also, the *camera coordinate systems* have to be logged, first, to take snapshots of relevant events or user interactions, and, second, to trace back camera pans. Previous approaches to interaction summaries were restricted to either the player's camera view [Halper and Masuch, 2003] or an overview camera view [Chittaro and Ieronutti, 2004; Hoobler et al., 2004]. More elaborate views would be desirable. Furthermore, *collision data* could be logged to illustrate scene modifications done by users. For example, object speed and direction could serve as input data to produce speed lines or similar visuals used in comic strips [Masuch et al., 1999; McCloud, 1994].

However, other technologies will require different visuals for emphasizing and tracing back the user's behavior and task accomplishment. Data specification, thus, is closely related to the specific implementation and visualization of the digital game-based training system.

Data Evaluation

Data evaluation has to analyze the assessed data as well as their according visuals. Based on the specified data, data evaluation has to select significant data. Hence, the evaluation is concerned with determining significance. Basically, the analysis has to answer two questions:

- Which data have to be included in the visual review log?
- How detailed should the visualization be?

Both questions aim at finding meaningful events that allow for a comprehensible review log visualization. Since the visual review log should be usable for classroom discussions, it should be readable. A straightforward solution to this problem would be to visualize any logged data and to restrict the data specified. This would make data evaluation unnecessary. However, data evaluation could be necessary for generating different visualization forms without specifying assessment and visual data anew. For example, coarse illustrations might only depict the course of the training session whereas detailed illustrations might even emphasize single events that are connected with some task accomplishment. Specifying data granularity should thus be contained in the reviewing component.

Data Visualization

Data visualization has to visualize and display assessment and visual data that were selected during data evaluation. Two visualization strategies have to be pursued:

- course-based visualization
- event-based visualization.

Course-based visualization concerns the visualization of the overall training course. Basically, data and events have to be arranged chronologically. Event-based data visualization concerns emphasizing specific events, e.g., highlighting of task or user interaction events that had been selected as relevant.

Course-Based Visualization Course-based visualization has to illustrate the overall course of events. Time data is essential for this as it could be used to arrange snapshots chronologically. Additionally, time could be encoded in single snapshots using color, size, transparency, etc. This could be useful to visualize a sequence of events, for instance. See Schumann and Müller [2000] for detailed information on time visualization and general information visualization.

Character positions are essential for deriving information on the character's movement by relating subsequent positions. Visual clues like dynamic glyphs [Nienhaus and Döllner, 2005] or storyboard elements [Goldman et al., 2006] could then serve to depict movement directions. Also, comic style elements as examined by McCloud [1994] and applied in [Gordon, 2006; Kurlander et al., 1996] seem useful for generating comprehensible visualizations of user behavior. Additionally, camera pans could be added, in case different camera systems were used for taking snapshots. Camera pan symbols could enhance the relationships between the different camera systems [Goldman et al., 2006].

Event-Based Visualization In contrast to course-based visualization, event-based visualization concerns the enhancement of single events. Non-photorealistic rendering techniques like stroke rendering and line drawings could be applied to enhance events or user interactions, for instance. Collision data could be used to identify the scene data to be enhanced.

Design and implementation of the reviewing component touches several distinct research domains. It is based on three mainstays: data specification, data evaluation, and data visualization. Users should have the possibility to modify each of these aspects according to their needs specifying different visualization styles or specifying different assessment data. The implementation of the reviewing component will, thus, be a challenging task for the developer.

4.6 Education by Playing?

Although the previous sections thoroughly discussed challenges and difficulties of designing digital game-based training systems, the question remains if education can be successfully supplemented by such applications. The goal of the following discussion therefore is to turn the question *education by playing?* into a statement saying *education by playing!*

In the context of digital game-based learning, one question constantly recurs: How much game supports learning? Even though the effectiveness of serious game playing to enhance the learning progress could not be verified so far [Lin et al., 2005; Wong et al., 2007], several studies showed the motivational effect triggered by computer and serious games in learning situations [Egenfeldt-Nielsen, 2005; Lin et al., 2005; Squire and Barab, 2004; Swing and Anderson, 2008; Wong et al., 2007]. At the very least, this makes serious games potentially useful for learning as educational supplements. The presentation of learning content as part of a game story and the combination of training and game playing strategies are two aspects that could neither be successfully implemented by edutainment applications nor by traditional learning environments that used game elements (cf. Sec. 2.3.2). However, serious games still have the reputation of supporting *playing in education* rather than *education by playing*.

To overcome this difficulty, this dissertation suggests further tightening the connection between education and playing by:

Integrating existing learning material (external data) into the game structure. Game players have to be constrained to use external data as part of game playing. In contrast to previous approaches, playing would become dependent on education, i.e., external material. Phrased differently, education would become an explicit basis for playing.

Addressing *all* participants of education as distinct game players according to their educational roles. Common educational situations have to be maintained and be pursued. In order to use new educational technologies, these have to be usable by teachers and students.

Reviewing the training sessions to obtain assessment. Student's training has to be assessed for the teacher's evaluation or self-evaluation. Moreover, classroom debate has to be prompted.

If all three aspects are integrated into digital game-based training systems, it can be assumed, in contrast to traditional serious games approaches, that

- teachers do not have to cope with completely new educational material and technologies
- students will be restricted in favoring gaming over learning and might more easily consider playing as learning support.

It will have to be proved, whether *education by playing* will finally be supported by implementations of the general framework. Yet, since the framework addresses central aspects of education and situates them in a game context, education by playing has been put onto a new level of digital game-based learning. In conclusion, a workflow schedule is provided next that will serve as implementation basis.

4.7 Workflow Schedule

Summarizing the discussion of design issues above, this section addresses the actual developers (system engineer, designer, ...) of a digital game-based training system. The section presents a workflow schedule that describes the implementation steps relevant to actually implement a digital game-based training system. The schedule can be divided into four major steps:

1. Establishing a user working profile:
   - examining user groups and educational situations
   - examining working material
   - choosing appropriate technology.

   The first step of the workflow schedule is to closely examine working processes of the user. The developer has to examine teacher and student user groups and get acquainted with teaching and learning situations. The working material, i.e., learning material, learning goals, and learning strategies, has to be evaluated next. Given such user-based information, a technology has to be recommended for implementing the digital game-based training system. At best it should supplement the specific educational situations, also in regard to user knowledge and educational meaning transfer.

2. Developing game-based training content:
   - applying the extended game universe
   - applying the general framework to determine the scope of the digital game-based training system.

   The second step of the workflow schedule is to design and specify the content of the digital game-based training system. This must be based on the user working profile and serve as a basis for the implementation. Therefore, the extended game universe has to be discussed in the context of learning material and game context; game story and player roles have to be developed. Finally, the scope must be clearly specified.

3. Implementing the game-based training system:
   - choosing the development tools
   - developing a data presentation
   - developing an authoring component
   - developing a training component
   - developing a reviewing component.

   The third step of the workflow schedule concerns the actual implementation of training content. Therefore, appropriate development tools have to be chosen first. For example, 2D/3D game engines are often times suitable to develop interactive 2D/3D environments because they are usually equipped with a wide range of functionality. Second, data presentation, authoring, training, and reviewing components have to be developed such that they suit the extended game universe. Especially, the implementation of authoring and training component has to support the general workflow of teacher and student.

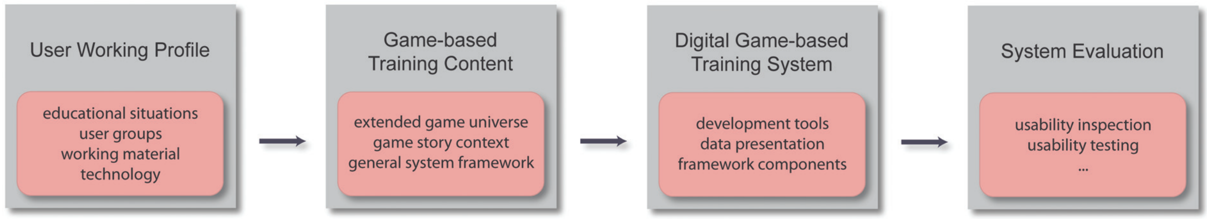


Figure 4.3.: The workflow schedule illustrates the four main steps of developing a digital game-based training system. First, background information has to be gathered to establish the user working profile. Second, game-based training content has to be conceptualized. Third, the digital game-based training system has to be implemented based on the previous concepts. Finally, system evaluation has to be regularly conducted to evaluate if the system suits user requirements.

4. Conducting system evaluation:
   - usability inspection
   - usability testing.

   The fourth and last step of the workflow schedule is to regularly conduct a system evaluation. Usability testing is a system evaluation method that can be conducted early on to ensure usefulness of the application. It will support communication between developers and users, which is essential for successfully developing a digital game-based training system. Further system evaluation methods like user studies should be conducted when the prototype is in an elaborate state.

The extended game universe should always be kept in mind at each stage, especially, when developing and implementing the game-based training content. Also, the primarily selected technology can change during the overall development process. In some cases, the technology might even be pre-determined by the user groups. In other cases, the developed game story might suit another technology than the one previously chosen. The schedule should be regarded as recommendation steps for the developer. Figure 4.3 illustrates the workflow schedule.

All schedule points have to be evaluated and discussed in close collaboration with the end users. Although this seems natural, in practice collaboration often is neglected due to time-constraints and tight resources. Still it is especially essential when users and developers come from different backgrounds. Lack of collaboration often results in misconceptions and tedious re-designing of software applications.

The workflow schedule also has to be considered in respect to current resources such as budget and manpower. Serious game development as well as development of other forms of digital game-based training systems usually lack support of a strong industry like the game industry [Martens et al., 2008]. For recent sales figures of computer games refer to [BIU, 2009; Entertainment Software Association, 2008; NPD Group Inc., 2009]). Research-based developments of digital game-based training systems should not try to compete with that sector in terms of photorealistic depiction, sound, etc. As Frazer et al. [2007] put it, “Less is more,” developers and designers should try to find simple and effective ways to transfer learning content.

The last section concisely presented relevant steps to develop and implement a digital game-based training system. These have been summarized as a workflow schedule. Naturally, the workflow schedule is related to instructional design methods like the ADDIE model. Yet it is particularly formed to the requirements of digital game-based training systems. It was used as a development and implementation basis for a prototypical digital game-based training system for crime scene investigation training called *OpenCrimeScene*. The next chapter describes how each workflow step has been applied to develop the digital game-based training system.

Chapter 5

OpenCrimeScene — A Digital Game-Based Training System for Crime Scene Investigation Training

This chapter presents the application of the general framework for digital game-based training systems to crime scene investigation (CSI) training. The collaboration project "Der virtuelle Tatort" between the Police College of Saxony–Anhalt, Germany, and the Otto–von–Guericke University of Magdeburg, Germany, served as a basis for this. The project goal was to develop a digital game-based training system that allows students to recapitulate criminalistic tactics and criminalistic techniques in regard to legal aspects. The digital game-based training system was intended to be used for lessons as well as for self-study. Reviewing was requested to assess the student's training progress. Software for desktop PCs was suggested. The context of the CSI training should address minor offenses like burglaries as a starting point.

The chapter describes the development of *OpenCrimeScene* and is structured according to the workflow schedule. The user working profile will be established first (cf. Sec 5.1). Second, the game-based training content of CSI will be set up (cf. Sec 5.2). Subsequently, the implementation of *OpenCrimeScene* will be presented. *OpenCrimeScene* contains several features necessary to support digital game-based training of crime scene investigation, as will be outlined in detail (cf. Sec 5.3). The chapter closes with system evaluation that has been conducted based on the final prototype of *OpenCrimeScene* (cf. Sec 5.4).

5.1 Establishing the User Working Profile

According to the workflow schedule, first of all the developer has to be acquainted with common working habits of the target user group. Therefore, the user working profile has to be set up in the context of CSI training. Regular visits at the Police College and discussions with colleagues from the Police College are necessary to establish the user working profile. Since education at the Police College of Saxony–Anhalt has already been discussed in Section 2.4, here a brief summary of user working habits will be given only.

5.1.1 User Group Specifics

The user groups that have to be addressed by *OpenCrimeScene* are police teachers and students. Most police teachers have already been working as investigators. They provide on-the-job expertise, which is beneficial when setting up realistic crime scene situations. In regard to software knowledge, teachers are typically familiar with text processing software like *Microsoft Office* applications. Yet use of computer games and computer graphics is not common at the Police College. Hence, police teachers are classified as *beginners* in the context of computer game software.

Students of the Police College usually are of at least 16 years and at most 25 years (middle civil services) or 28 years (upper civil services). In contrast to police teachers, the user group of police students can be considered “digital natives” because they most probably have grown up playing computer games. They can be classified as *experienced* game players. To further examine working habits of both groups, educational situations have to be considered.

5.1.2 Educational Situations

In the context of *OpenCrimeScene*, practical training situations at the Police College’s exterior and interior crime scenes are of most interest. In regard to the educational universe, these situations split into three categories (cf. Sec. 3.1.1):

Teaching situations comprise teaching of criminalistic tactics and criminalistic techniques to protect and assess the practice crime scenes of the Police College (cf. Section 2.4.2). The police teachers recreate crimes like burglaries or accidents and, in doing so, prepare the crime scene as authentically as possible. On-the-job expertise contributes to reproduce realistic crimes.

Learning situations comprise acquisition of how to conduct criminalistic procedures. These practical training situations require the students to have already studied how to protect and assess the scene in theory. Specific criminalistic techniques like collecting and processing evidence can be studied in the Police College's crime lab.

Communication situations police teachers and students depends on the specific training situation. Generally, teachers prepare the tasks that students have to accomplish, assess the student's behavior, and discuss problems like in most educational situations.

What makes these educational situations stand out is that they require several practical CSI tools, so-called forensic tools to protect the crime scene. Hence, an acquaintance with the specifics of CSI working material is essential to develop an authentic digital game-based training system for CSI training.

5.1.3 Working Material Specifics

Police students have to train how to protect the crime scene. This comprises training of criminalistic procedures. Thereby students have to train how to handle CSI tools like, for example, brushes and powders to collect fingerprints. Figure 5.1 illustrates a subset of tools used for crime scene investigation. Such tools must be integrated into *OpenCrimeScene* in order to reproduce training procedures authentically.

In addition to technical working material, police students have to know about law and the legal basis upon which the investigation is established. For instance, imminent danger (in German "Gefahr im Verzug") overrules the legal restriction of a search warrant and allows investigators to act immediately. It is, hence, necessary to train different legal scenarios and to test the student's knowledge on such issues. Quizzes and context-dependent help should therefore be integrated into the prototype.

Apart from supporting the training point of view, the extended game universe is also intended to support the teaching point of view. This means, "working material" used by police teachers to reproduce crimes has to be present in the digital game-based training system. Such tools could be all kinds of tools used by burglars or offenders like jimmies, hammers, etc.

Figure 5.1.: Forensic tool box use for crime scene investigation. A subset of forensic tools is depicted above (from left to right): chalk, shears, spoon, different brushes, powders, gas, tweezer, magnifier, foil, boxes. Images courtesy of PHK Peter Eichardt, Police College of Saxony–Anhalt, Germany.

At this point it is worthwhile noticing the close link between CSI education and the general framework caused by the integration of such "offender tools." Police teachers reproduce housebreaking by using such tools and leave tool marks at the crime scene. Police students have to train how to assess such tool marks because they might lead back to the tool's owner. Hence, the integration of "offender tools" not only would serve as authoring element but also as a connection between authoring and training or, phrased differently, would strengthen the dependency of back-story and main-story. Yet before the development of game-based training content will be discussed in Section 5.2, a quick glance has to be taken at the technology used for *OpenCrimeScene*.

5.1.4 Technology

The Police College requested a technology that could easily be used by every student and teacher. Desktop PCs seem to be of particular suitability for this because it can be assumed that most students own or have access to a PC. Also, purchase of a set of PCs for educational purposes at the Police College can also be assumed as reasonable. Furthermore, recent desktop PC hardware provides a stable technical foundation to execute and work with interactive 3D graphics environments. Thus, *OpenCrimeScene* has to be implemented as standalone application for desktop PCs.

5.2 Developing Game-Based Training Content

After having examined common user working habits and having set up the user working profile, the game-based training content can be developed on this basis. To ensure a consistent relationship between CSI training at the Police College and digital game-based CSI training, the extended game universe has to serve as root position (cf. Chap. 3). The relationship will be described below.

5.2.1 Extended Game Universe in CSI Training

The extended game universe contains six elements that have been determined by the educational universe (cf. Tab. 3.2). If applied to CSI training, this results in:

1. the game (story) has to be determined by CSI learning material
2. the game player roles have to be determined by police teacher and police student
3. the relationships of problem presentation and problem solving are determined by preparing a crime scene and investigating a crime scene
4. the relationship of teacher-student-communication has to be supported by reviewing criminalistic procedures.

Central to implementing these elements with regard to *OpenCrimeScene* is the creation of the game context (cf. Sec. 3.2.1). Since story creation has been discussed as one main aspect of game context creation, learning material and user groups will be directly related to game story and game player roles.

Establishing Game Story & Game Player Roles

Practical training at the Police College's crime scenes comprises two aspects:

- recreating a crime, which is done by the teachers

- investigating a crime, which is done by the students.

Crime recreation and investigation correspond to *problem posing* and *problem solving*. Both situations serve well for developing a "cops & robbers" game story; the overall conflict being good vs. bad. In such a story setting, CSI working material can nicely and authentically be added for cops and robbers each. Cops have to be played by the students and robbers have to be played by the teachers. Both roles have to provide equal human properties such as leaving fingerprints when touching some scene object or leaving footprints when walking through the environment. This is necessary because, first, students have to learn how to collect physical evidence and, second, students have to learn how to prevent scene contamination with their own evidence. This requires designing the interaction means for both roles authentically. For instance, the robber role has to be provided with interaction means to destroy and remove inventory, whereas the cop role has to be provided with features to collect evidence and protect the crime scene.

Since *OpenCrimeScene* is a pilot project in terms of collaboration and project goal, and since resources of time, money, and developers are truly limited, the storyline has to remain straightforward and understandable. The main goal is to prove the prototype's applicability to CSI education first of all. Regarding the story theme, the cops & robbers game will be put into the context of housebreaking and, thus, introduce minor offenses to the users. In that sense, back-story and main-story are determined by breaking into houses (back-story) and investigating housebreaking (main-story). The conflict of the back-story is to get away with the crime, whereas the conflict of the main-story is to solve the crime and arrest the criminal. This way, both stories are closely dependent on each other. The implementation of the sub-stories as well as the assessment of the training have to be implemented by the framework components.

Establishing Authoring, Training, & Reviewing

As was already discussed, recreating a crime and crime investigation correspond to problem presentation and problem solving and requires making available certain interaction techniques for the players each. Furthermore, the specifics of housebreaking determined when playing the back-story have to serve as input for the specifics of the investigation of the main-story. The dependency of both sub-stories relates "cops & robbers" as well as students and teachers implicitly. To explicitly link both story situations to education, they have to be assessed and be summarized as a visual review log. This has to be

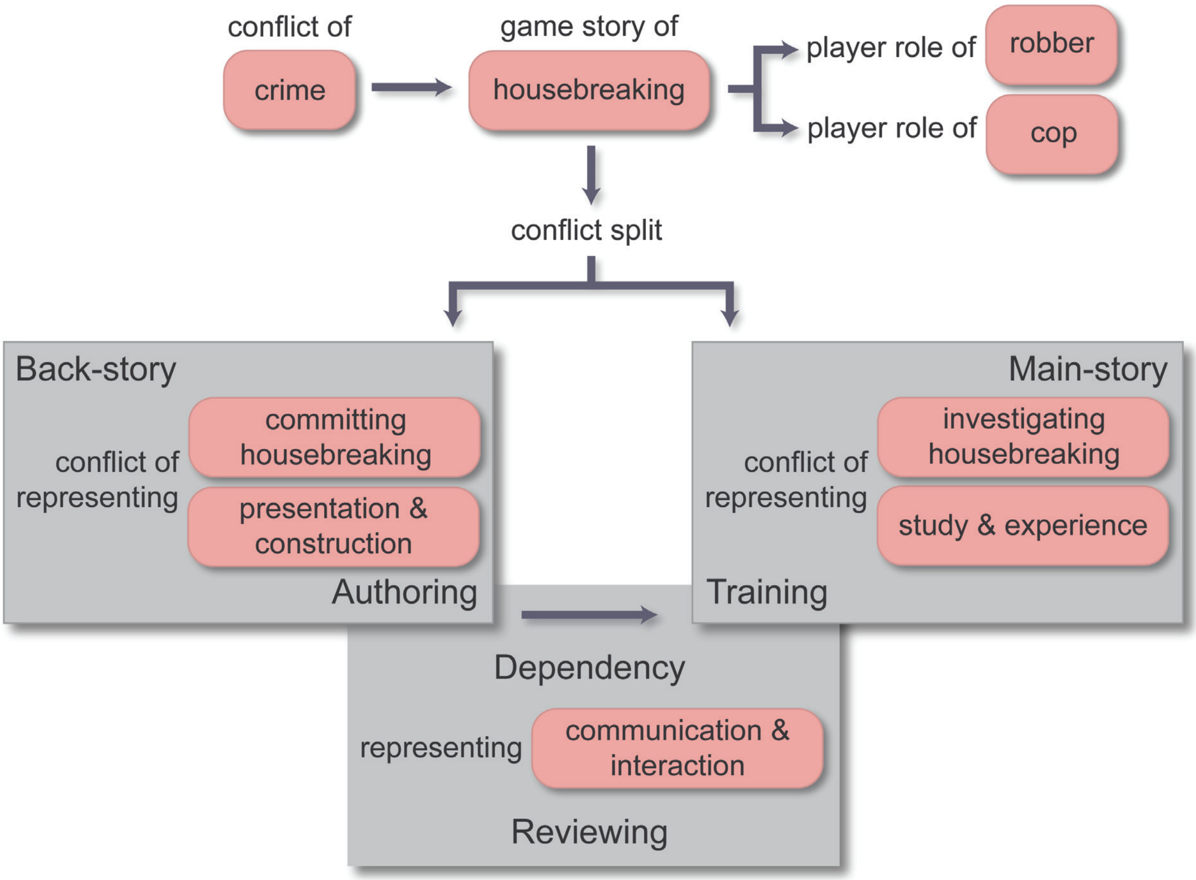


Figure 5.2.: Creation of the game context for *OpenCrimeScene*. The conflict of a *crime* is embedded in the story context of *housebreaking*. Two player roles of *robber* and *cop* are introduced, accordingly. Based on the game story, the conflict is split into *committing housebreaking* and *investigating housebreaking*, which represent the conflict of back-story and main-story, respectively. Regular educational situations at the Police College relate to this scenario. Also, both stories depend on each other and allow for fostering the relationship of communication & interaction by reviewing training sessions.

implemented in the reviewing component. Therefore, the reviewing component has to log typical procedure of authoring and training in order to assess how successful training has been. The visual review log should then depict special events that occurred during training like, for example, correct or false behavior of the student. Following the illustration of the game context creation depicted in Figure 3.2, Figure 5.2 illustrates the creation of game-based training content for *OpenCrimeScene*.

In accordance to the extended game universe, game story content has been developed in the last sections. Subsequently, the actual scope of the prototype has to be specified with regard to the general framework. This will be described in the next section.

5.2.2 General Framework & Prototype Scope

In general, the development of game-based training content is divided into two categories: application of the extended game universe to the specific learning content, that is, CSI training in this case, and application of the general system framework to the prototypical system along with specification of the prototype's scope. The former category allows the developer to freely create ideas of story content, story elements, and story roles that represent the educational context. This is important as it allows for examining the educational context in different game story situations and, thus, literally supports the idea of game-based training. In a second step, it is of equal importance to restrict the set of ideas and created game elements to those elements essential for the prototype and to clearly define the prototype's scope. This is especially relevant in the context of project work because resources are generally limited yet an end product is required nonetheless.

To specify the scope of a prototypical digital game-based training system, the general framework has to serve as the root position. In the context of *OpenCrimeScene*, the prototype was restricted to a basic set of game data and CSI training functionality. The primary goal was to prove the system's applicability to CSI education. In regard to the general framework, the scope of *OpenCrimeScene* is comprised of the following elements:

Data presentation has to be comprised of

- Internal game story data like a virtual house or room.
- Internal player-specific data like
  - player characters for cop and robber
  - forensic tools
  - physical evidence like fingerprints, footprints, and blood traces
- External data like learning material on law and legal issues to prepare quizzes and context-dependent help.

The authoring component has to allow the teacher to play the back-story and prepare the crime scene by

- destroying, modifying, and removing inventory
- leaving evidence
- preparing a quiz or help menu.

The training component has to allow the student to play the main-story and investigate the crime scene by

- taking pictures of the crime scene
- finding and collecting evidence
- answering questions or accessing help menu.

The reviewing component has to allow for assessing how the student investigated the housebreaking.

The presented scope does not contain all elements that were discussed during the design phase like “offender tools,” for example. However, the first prototype was intended to put more weight on the training part of the digital game-based training system. The subsequent sections will now discuss development details and the implementation of *OpenCrimeScene*.

5.3 Implementing the Training System

The implementation of the first prototype was conducted from September 2005 to September 2007 at the Otto–von–Guericke University of Magdeburg. Even though the previously introduced scope of the digital game-based training system consists of a manageable set of elements, each of them had to be newly developed, modeled, and implemented. This could not have been done without the contribution of about 18 students who added to the implementation of *OpenCrimeScene* in the course of seminar assignments, diploma theses, or as student assistants. Their input was supervised by the author of this dissertation and will be mentioned where appropriate.

Prior to presenting implementation details of *OpenCrimeScene*, a few general remarks seem appropriate at this point to better establish the following. In general, the development of the game-based training content described in Section 5.2 serves as an intermediate between the users and their working habits on the one hand and the implementation of the training system on the other hand. This whole procedure, summarized under the name of workflow schedule, should not be considered as a linear sequence but rather as a circular structure. In that sense, implementation provides a link back to users and emphasizes user-centered development, which was identified as the central design issue in Section 4.1.

In this regard, the following subdivisions present implementation details of *OpenCrimeScene* by taking into account the established user working profile. Starting with the development tools employed, the subsequent sections correspond to the general framework structure and are split into data presentation, authoring, training, and reviewing components. A brief system overview finally outlines *OpenCrimeScene*. The general development of *OpenCrimeScene* was presented and published in [Brennecke et al., 2008b]. Specific details on individual elements of the system as well as related aspects to support certain rendering techniques have been published in [Brennecke et al., 2008a, 2007; Isenberg and Brennecke, 2006; Isenberg et al., 2005].

5.3.1 Development Tools

According to the user working profile, *OpenCrimeScene* should be implemented as an interactive 3D graphics environment for desktop PCs. Yet implementation of such environments is costly in most cases because it requires different sets of functionality like 3D graphics, user input control, physics, sound, networking, collision detection, etc. A so-called *game engine* is a compound of numerous such libraries and serves well for implementing interactive 3D graphics environments. In addition, several engines are equipped with an integrated level editor that allows for designing game scenarios and levels.

Consequently, implementation of *OpenCrimeScene* was conducted using the open-source 3D game engine *Delta3D*, version 1.2.0.[1] Different engines have been considered as well, for example, *Shark3D* engine.[2] However, *Delta3D* provides the most features: numerous well-known open-source libraries are employed to supply extensive core functionality. Among others, *Delta3D* combines *OpenSceneGraph*, *OpenAL*, *ODE*, *Game Networking Engine*, and *Qt*.[3] In addition, *Delta3D* uses the *Python* language to support scripting. It is equipped, furthermore, with several useful design tools like a level editor or a graphical particle effect editor. Most crucially, however, the engine is explicitly designed to develop training and simulation games. The downside of *Delta3D*, however, is its frequently changing development state and functionality due to the compound of different libraries. Nevertheless, it seems to be an appropriate development tool, especially in regard to

1 For further information see [http://www.delta3d.org] (March 22, 2009).

2 For further information see [http://www.spinor.com] (March 22, 2009).

3 See a complete overview at [http://www.delta3d.org/article.php?story=20050707151113592&topic=docs] (March 22, 2009).

future implementations of *OpenCrimeScene*. The variety of development possibilities has been the decisive factor to use *Delta3D*.

Apart from the implementation tool, *OpenCrimeScene* also requires application of several image and model processing tools like *Adobe®Photoshop®* to create textures, *3ds Max* and *Maya* to create 3D models and animations. These have to be used, particularly, to create and present game data, as will be described in the next section.

5.3.2 Data Presentation

Data presentation is an essential part of a digital game-based training system. A lot of work is concerned with basic development tasks like modeling and texturing scene objects, or designing graphical user interfaces. In addition to such basic tasks, novel concepts have to be developed to present the specific learning content in an interactive 3D environment. In this context, authentic data depiction as well as authentic data behavior are major challenges to a digital game-based training system and clearly distinguish digital game-based training systems from computer games. The computer game *CSI Miami* addresses crime scene investigation and realistically depicts evidence in the crime scene. However, data behavior as well as the conduct of criminalistic procedures is provided by the game on a rather superficial level.

Since *OpenCrimeScene* is an interactive 3D environment, scene objects should be represented as 3D representations to allow users to interact with such objects like in real life and to strengthen the link between virtual reality and reality. In contrast, graphical user interface elements should be represented as 2D representations to support clarity and to abstract from the 3D scene. Furthermore, data storage and access, especially of general game story data, should be provided using the concept of *game level*. Each level represents a case file and contains the data necessary to play the specific case.

As proposed by the general framework, player-specific data has to be stored in the corresponding authoring and training components (cf. Sec. 3.2.2). Since some of the player-specific data can only be discussed in the context of authoring or training to support comprehensibility, the following subsections present data development of game story data, player characters, and graphical user interfaces. External data integration will be briefly discussed subsequently. Development of further player-specific data like different types of evidence or forensic tools will be discussed in the corresponding authoring or training components, respectively.

(a) (b) (c)

Figure 5.3.: The virtual crime scenes. Figures (a) and (b) adapt the interior practice crime scenes, whereas (c) adapts the exterior crime scenes. Models (a) and (c) courtesy of Reinemann and Hähnel [2006].

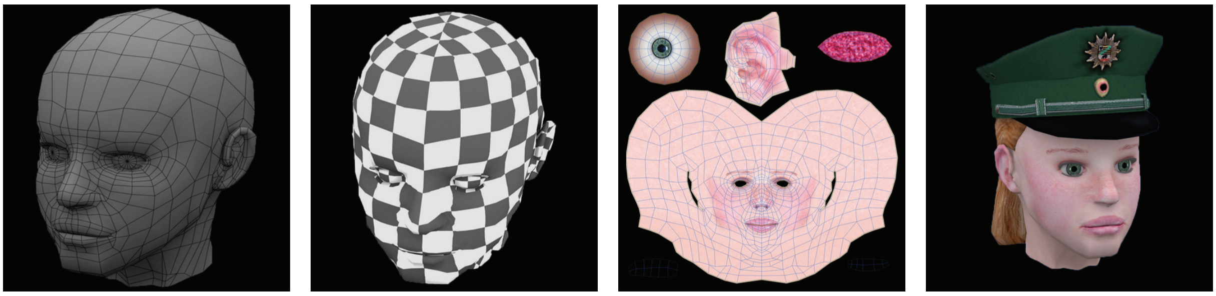
Virtual Crime Scene

The virtual crime scene has to become the central element of *OpenCrimeScene*. It literally connects back-story to main-story and relates robber to cop and teacher to student, respectively. Thus, authentic presentation is essential. Also, to explicitly link virtual training to real training, the virtual crime scene has to be modeled on the basis of the practice crime scene at the Police College. The virtual crime scene modeled for *OpenCrimeScene* basically reproduces the practice scenes, as illustrated in Figure 5.3.

Different crime scenes can be created using *Delta3D*'s level editor *STAGE*. Apart from arranging 3D models in the environment, the editor can be used to assign game properties to the scene objects. Such properties support the realistic rendering of the objects and can be used in the source code to modify the objects.

Ideally, all scene objects should be assigned physical behavior and collision response functionality. However, during the time spent developing *OpenCrimeScene*, both features were not supported properly by the game engine. As a consequence, scene objects have to be assigned general transformation properties to reproduce physical behavior. User interaction with scene objects then can result in necessary object transformations. Although this is only a workaround that does not comprise automatic scene object behavior, it at least allows for emulating realistic data behavior of the first prototype. Also, recent versions of *Delta3D* will have solved those difficulties.

Each crime scene is stored in a game level and can be accessed by the different components. To explore crime scenes, player characters have to be developed, as will be described next.



(a) (b) (c) (d)

Figure 5.4.: The sequence of images illustrates how to develop a character's head. First, the 3D model has to be created (a) and texture coordinates be assigned to it (b). Then texture images have to be generated that fit to the texture coordinates (c). Finally, the character's head can be created by mapping the textures to the according coordinate values (d). Images courtesy of Schulze and Maertins [2006].

Cop & Robber Characters

Cop and robber characters are part of player-specific data. To minimize the overall development cost, however, it seems to be useful to model one character and to adapt its look to either the cop or robber role. In that sense, the cop character has to be dressed in a realistic-looking police uniform, whereas the robber character's clothing can be chosen freely. The characters were modeled in the course of a student assignment by Schulze and Maertins [2006].

Basically, character development requires three steps:

1. Creation of a 3D character mesh.
2. Visualization of the character mesh using texture images.
3. Generation of character animations using animation and skeletal data.

To create the character mesh, an authoring system like *3d Max* can be used. It provides extensive features to model and animate 3D object meshes. Given a character mesh, human features have to be added. This is normally done with texture images, which are mapped onto the object mesh. Texture mapping, however, requires generating texture images that fit the object mesh without being distorted. Therefore, texture coordinates have to be generated for each of the object's vertexes. According to the texture coordinates, texture images can then be painted. Figures 5.4(a) to 5.4(d) illustrates the character's head with according character mesh, texture coordinates, and texture images.

Figure 5.5.: The upper sequence of images illustrates photographs of a real police uniform. Images courtesy of the Police College of Saxony–Anhalt, Germany. The lower sequence of images illustrates the textures that were created based on these images for the cop character of *OpenCrimeScene*. Images courtesy of Schulze and Maertins [2006].

To realistically generate textures of the police officers uniforms, the Police College provided a large set of photographs. These served as a basis to develop the cop's clothes (see Fig. 5.5). For simplicity, black texture was chosen to depict the robber character.

Once the character is modeled and textured, animation can be added. Therefore, animation and skeletal data are required. To reduce development cost, these were taken from the Carnegie Mellon University Motion Capture Database. Data are freely available and can be accessed online.[4]

To create an animated character, first of all, skeletal data have to be animated. Then, the animated skeleton has to be assigned to the textured 3D mesh. Since *3D Studio Max 8.0* could not properly process the skeletal data, *Maya 6.0* was employed to animate the skeleton and assign it to the 3D character. The final result is depicted in Figure 5.6.

To integrate the character into *OpenCrimeScene*, the character animation libraries provided by *Delta3D Cal3D*[5] and *ReplicantBody*[6] can be used. They allow for referencing the character and its animations in the source code. This way, user control can be mapped to different character animations. For example, the character's animation to turn left will

4 For further information see [http://mocap.cs.cmu.edu/] (March 22, 2009).

5 For further information see [https://gna.org/projects/cal3d/] (March 22, 2009).

6 For further information see [http://www.vrlab.umu.se/research/replicants/] (March 22, 2009).

(a) (b) (c)

Figure 5.6.: The images illustrate the character in default position seen from front (a) and back (b). Image (c) illustrates the character's walking animation. Images courtesy of Schulze and Maertins [2006].

be executed, if the user presses a certain key, etc. The following section on graphical user interfaces goes into that in more detail.

Graphical User Interfaces

So far, geometric data have been described. To access and interact with such data, graphical user interfaces are required. Generally, such interfaces are elaborately designed in most computer and professional serious games. To meet this standard, a collaboration project was conducted between the Department of Simulation and Graphics of the University of Magdeburg and the Industrial Design Institute of the University of Applied Sciences Magdeburg-Stendal during the winter semester 2006/2007. The goal of the collaboration was to design system menus as well as to develop a general scheme for user interaction. The results have been partially integrated into *OpenCrimeScene*, and partially been developed as demo programs using Macromedia Flash and Director to illustrate the approaches. Details can be found in [Grubert and Habener, 2007] and [Reinke and Seidl, 2007].

General Considerations In *OpenCrimeScene*, user interface design has to underpin virtual criminalistic procedures. Furthermore, it has to support non-experts and appeal to experienced game players. Both issues require examination of criminalistic procedures and current standards of user interface design employed in recent computer games. The interface design is based upon the examination.

In a digital game-based training system like *OpenCrimeScene*, users have to explore the 3D scenario, coordinate and apply different tools, access case files, solve tasks, etc. To classify user interaction, two basic interaction modes will be introduced here:

Exploration mode summarizes user interactions in a 3D scene. Since navigation and object interaction in a 3D interactive environment is non-trivial, especially, for non-expert users, two exploration modes seem appropriate here:

Navigation mode describes user interactions during navigation. It adapts from one of the most prominent player-control-paradigms in computer game playing introduced for *Quake*: the user player can navigate through the environment using the WASD-keys of the keyboard in conjunction with the mouse. The keys can be used to walk forward (W), backward (S), sidestep left (A), or right (D), whereas the mouse can be used to control the viewing direction. As was already mentioned in the previous section, the walking keys are linked to the character's walking animation. Furthermore, the player can select scene objects when using the CTRL-key and *picking* functionality.[7]

Editing mode describes special user interactions with scene objects like, for example, to process evidence. It uses a different representation for such interactions: the user can select a scene object in the virtual crime scene via picking. Then, to interact with the object the user has to switch to editing mode, which is outside of the virtual crime scene and supports to better interact with the scene object. In editing mode the user can not navigate but can transform and modify the selected object. Mouse and keyboard control for either navigation or editing mode can be freely configured by the individual users. Hence, users are not restricted to, for example, WASD-keys to control navigation but can specify their own input control. A configuration file can be adapted to individual preferences in order to further simplify application use.[8]

GUI mode describes user interaction with system menus. These are 2D representations and split into three menu types: a general system menu, a back-story menu, and a

7 Different solutions to implement picking exist, for example, whenever the user selects or "picks" a scene object using mouse and cursor, a ray is cast into the scene and the intersection of the ray with the scene object is calculated.

8 Thanks to Jens Grubert for his effort in implementing individual input control configuration.

main-story menu. The main challenge of designing such user interface menus is to support users when interacting with the system. Menu items should be designed clearly and represent their functionality in an unambiguous fashion in order not to distract the user. Each of the menu types developed will be examined separately below.

General System Menu

The general system menu is responsible for general game administration. Basically, it has to allow the user to select different game levels and to grant access according to whether the user is teacher or student. In addition, training progress should be recognizable. To do so, the notion of *case files* seems an appropriate metaphor to depict different game levels in CSI training. Moreover, *time* seems another appropriate metaphor to depict training progress. It supports clear structure of different game levels; users can easily recognize how far the training has advanced. Two alternative design solutions were developed, as illustrated in Figure 5.7.

The first solution, illustrated in Figures 5.7(a) to 5.7(c), quite naturally implemented the idea of case files. Different case files can be accessed via a filing cabinet. Two drawers grant access either to teacher or student. In the respective drawer, the files are chronologically ordered and can be viewed when selected. The second solution, illustrated in Figures 5.7(d) and 5.7(e), focused on the time aspect of training and distribute the cases according to a time line on a plain. When selected, the case file can be opened and presents a snapshot of the respective 3D scenario.

Although both solutions are professionally and beautifully designed, the first solution was integrated into *OpenCrimeScene*. This decision was made because the first solution seemed to be more straightforward and clear in regard to non-expert users.

Back-Story Menu

Design of the back-story menu has to support non-expert users in preparing the crime scene and playing the robber. Especially, the back-story menu has to support explicit authoring. Therefore, the teacher has to be given access to different types of evidence or robber tools to modify the crime scene. Furthermore, the back-story menu should allow the teacher to create tasks or help texts. However, since the focus of the first prototype of *OpenCrimeScene* was put on the training aspect, the back-story menu was only conceptualized in the course of the collaboration project.

The concept included a procedure to add evidence to the crime scene using editing mode. This is illustrated and briefly described in Figure 5.8.

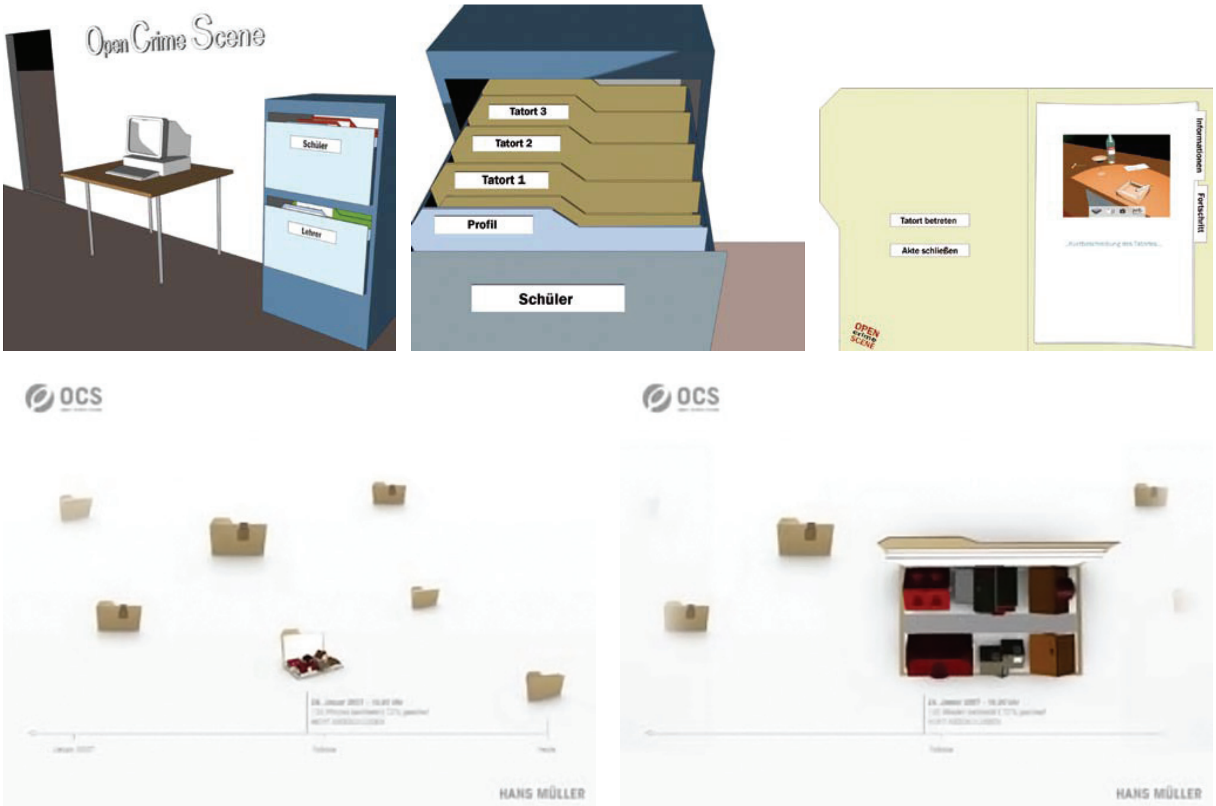


Figure 5.7.: The images illustrate the different solutions of the general system menu. The upper sequence of images naturally adapts the idea of case files and uses file cabinets to depict different game levels. Drawers can be opened either by teacher or student players. Images courtesy of Grubert and Habener [2007]. The lower sequence of images focuses on the time metaphor and mapped the case files on a time line. Moreover, the relation between case file and game level is strengthened: when selecting a case file it shows the according game level. Images courtesy of Reinke and Seidl [2007].

Main-Story Menu Design of the main-story menu has to support users when investigating the crime scene and playing the cop. Thus, the main-story menu has to support the training aspect of *OpenCrimeScene*. The student has to be given access to different types of forensic tools and CSI tools to protect and assess the crime scene. Also, the main-story menu has to allow the student to accomplish prepared tasks and access help texts. Since *OpenCrimeScene* primarily focused on the training aspect in the first prototype, two different solutions were developed for main-story menu design. Figures 5.9 and 5.10 illustrate the solutions.

Both solutions focus on presenting forensic tools because this was of primary interest to support the user when playing the main-story. The first solution provides a bottom menu that offers access to the tool box and an SLR camera. The second solution also provided a set of forensic tools. Additionally, the second solution developed a representation for editing mode similar to the back-story menu. When switching into editing mode, the user

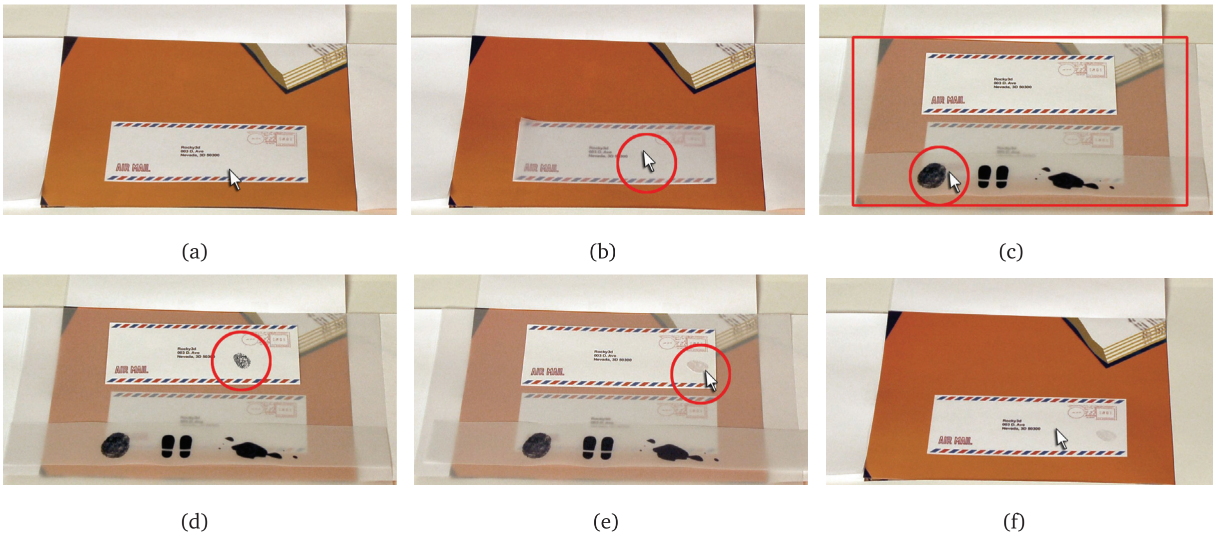


Figure 5.8.: Concept of the back-story menu. If the teacher player selects an object in the virtual crime scene, the system has to switch from *navigation mode* in the scene to *editing mode*. In editing mode, the player has access to different types of evidence that can be placed on the object. After leaving editing mode, evidence has been added to the scene object and the player can continue navigating through the scene. Images courtesy of Reinke and Seidl [2007].

can select different evidence identifiers and process evidence. To ensure that navigation and editing mode are put into context, the representation of editing mode is designed as an additional and transparent layer put in front of the 3D environment. This way, both modes harmonize with each other and provide consistency.

So far, the first solution has been integrated into *OpenCrimeScene*, as illustrated in Figure 5.11. Forensic tools can be selected by the user. Use of the forensic tools will be described in the according section on the training component. However, integration of an additional layer for editing mode is still due because of time restrictions. Therefore, a combination of both solutions will be necessary. Moreover, design of task accomplishment will be part of future work due to the same reason of tight time resources.

The previous sections gave a first insight into implementation details on internal data. Further details on player-specific data are given in the subsequent sections. Before going into that, a brief overview of external data is presented next.

External Data

As was mentioned in Section 5.1.3, in addition to practical training and handling of CSI tools, police students have to know about the legal basis upon which they investigate.

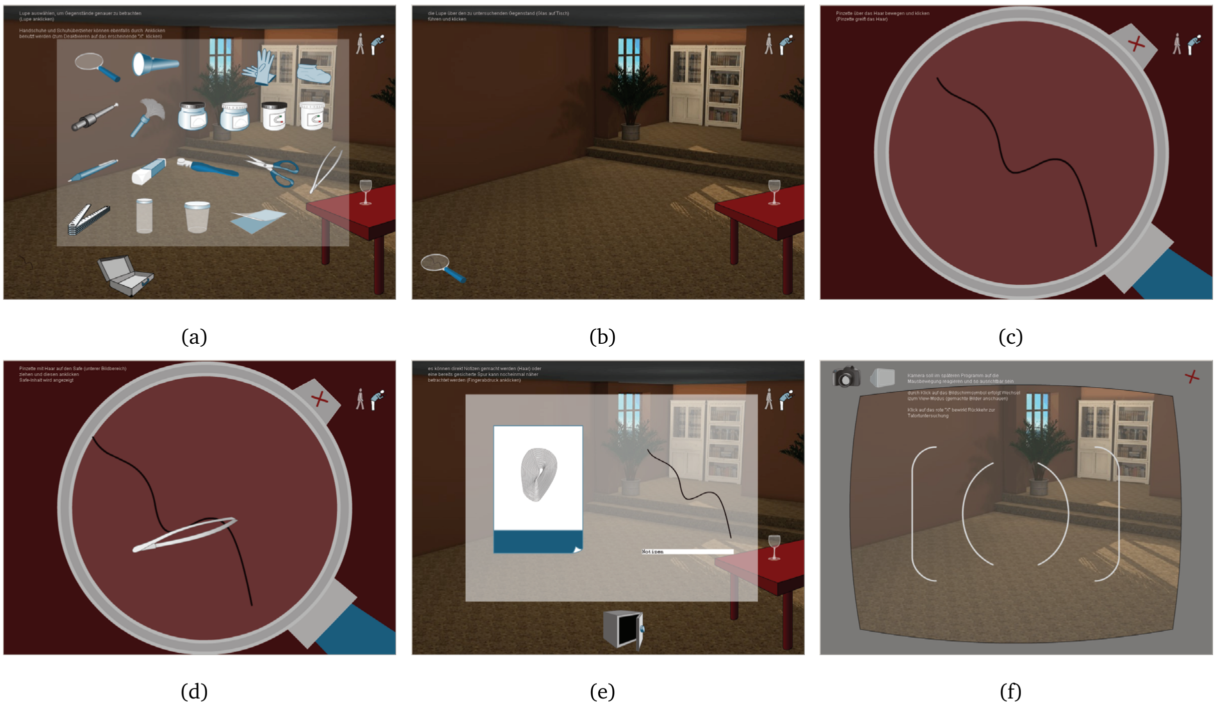


Figure 5.9.: This solution of the main-story menu uses a cartoon-based style to depict the different forensic tools. In the demo of this solution, the user can select the magnifier glass (a), which becomes the cursor icon and can be used to select a hair (b). The editing mode is editing as a result of the selection (c). After having selected the tweezers from the tool box, the user can add the hair to the evidence box (d). Finally, the user can also select an virtual SLR camera and take a snapshot of the scene (f). Images courtesy of Grubert and Habener [2007].

Quizzes on legal issues suit the demand to explicitly connect theory and practical training, which is currently not provided for by the Police College. One goal of *OpenCrimeScene* is to offer teachers a way to prepare such quizzes during the authoring process by using existing learning material. This way, teachers could adapt training sessions to current lessons, which is a central goal of the general system framework (cf. Sec. 3.1).

Integration of existing learning materials requires special system interfaces to represent such external data in the training system. In terms of quizzes, external data could be text material, images, or even video sequences to illustrate the quizzes. Integration of such data requires developing three different types of wrapper interfaces that can display and process text, image, and video formats in the virtual scene. Such wrapper interfaces have neither been designed nor developed for the first prototype though. Their integration will be part of future work and, as such, will be briefly discussed in Section 6.2.

However, in the course of his diploma thesis, Spengler [2006] developed a concept for an interactive quiz that adapts to the current legal basis. The quiz can be established by

(a) (b) (c)

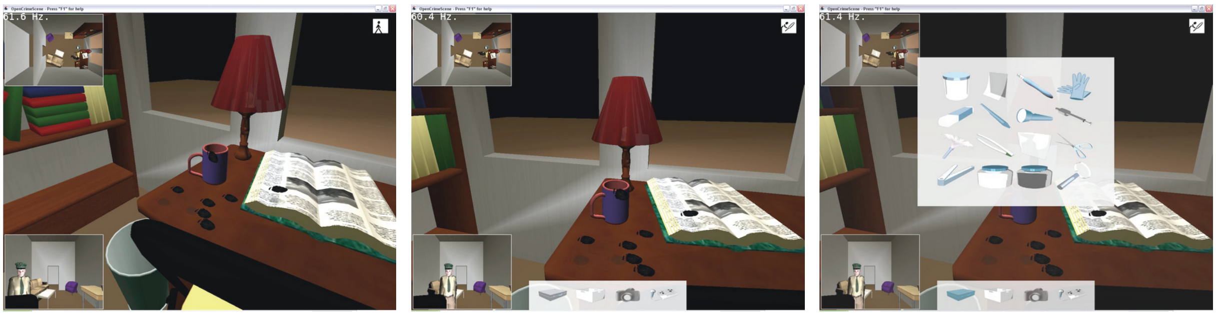
(d) (e) (f)

Figure 5.10.: This solution of the main-story menu makes use of an additional layer to distinguish between navigation mode (a) and editing mode (b). The image sequence depicts how to secure a fingerprint in editing mode. Therefore, a set of forensic tools is presented to the user on the lower part of the screen. To secure the print, it has to be recognized first, which can be achieved by using the torch tool (c). Then, a brush has to be selected and applied to the fingerprint (d), which becomes visible after dusting (e). Finally, it can be taped and stored in an evidence box. Images courtesy of Reinke and Seidl [2007].

the teacher as part of explicit authoring. The concept will be discussed in the context of the authoring component in the subsequent section.

5.3.3 Authoring Component

The authoring component is intended to support the police teacher in adapting training sessions to current lessons by acting as a robber and playing the back-story of housebreaking. Therefore, in-game authoring has to be implemented (cf. Sec. 3.2, esp. Sec. 3.2.2). This involves developing techniques to implicitly commit housebreaking and leave evidence traces at the crime scene as well as to explicitly prepare the crime scene with evidence traces and questions. A realistic presentation of trace data that is especially important in regard to the educational purpose of *OpenCrimeScene* because police students have to learn how to identify and assess evidence traces.



(a) (b) (c)

Figure 5.11.: The images show the main-story menu that was integrated to *OpenCrimeScene* by Grubert and Habener [2007]. Navigation mode is visually expressed by a *walking icon* in the upper right corner of the screen (a). When switching to the editing mode, the icon changes to *investigation icon* (b). Via the bottom menu, the user can select tool box, evidence box, SLR camera, or microphone. The tool box contains forensic tools (c).

Presenting Evidence Trace Data

Evidence types typically split into

- physical evidence
- documentary evidence.

Physical evidence is any kind of physical object that contributes to reconstruct a crime. Such evidences can be human traces, textiles, technical impressions, biological traces, and many more. In contrast, documentary evidence is produced by investigators to document physical evidence found at the crime scene. Such evidences are photographs, notes, video sequences, etc. Since documentary evidence is produced during investigation, it will be discussed in the context of the training component as a result of forensic tool use (cf. Sec. 5.3.4). In this section, special emphasis is placed on physical evidence caused by human-object-interaction. Due to their uniqueness as identification means, they are given high priority in police education. Consequently, they had to be integrated into *OpenCrimeScene* prior to other physical evidence traces.

Human Traces In general, physical evidence is defined as visible or latent alteration of material found at the crime scene. In most cases, such material alterations occur after one or more objects interacted with each other. In the case of human evidence, humans transfer fingerprints when touching objects, blood traces when getting cut by an object, footprints when walking on the ground, etc. The type of interaction as well as the objects involved determine how the traces appear in the crime scene: the geometric distribution of human traces depends on the kind of interaction that was conducted. The visual appearance of human traces depends on the material of the object involved. For example, blood drops, smear, or splashes result from different interactions; fingerprints typically are latent on opaque material and visible on transparent material.

In addition, human traces might change over time due to their own physical properties or due to external influences. For example, blood turns from red to dark red or even brown over time; rain might wash away blood traces and footprints or modify their visual appearance such that they might not be recognized as bits of evidence anymore. Consequently, visual appearance and geometric distribution of real traces depend on four aspects:

- physical properties of trace data themselves
- physical properties of scene objects and environment (including lighting conditions)
- interactions that caused trace transfer
- following alterations of trace data caused by interaction.

Realistic presentation of human trace data in a virtual environment that is based on these aspects is challenging and can easily become highly complex. In any case, it requires specifying a representation for trace data first of all. Use of texture images would provide a straightforward solution for this. Next a simulation model would be required that describes different types of interaction between player character and scene object. Among others, the simulation has to specify which part of the object was touched by the player character and how the interaction was conducted. For example, in the case of “grabbing,” it has to specify which hand was used, which part of the hand touched the scene object, how the scene object was touched, etc. Figure 5.12 illustrates different types of fingerprint and handprint data that contribute to the reconstruction of the criminal’s behavior.

According to these specifications, trace data would then have to be fully or partially distributed over the scene object. In the case of texture images, this requires specifying texture coordinates for each scene object to provide individual textures for each object component in order to realistically blend trace texture with the scene object texture and to map trace texture onto the specified area of the scene object without distorting it. A realistic visualization could be achieved by using shader programs to blend both textures according to their physical properties and current lighting conditions. As can be seen,

(a) (b) (c) (d)

Figure 5.12.: The images illustrate a set of real fingerprints investigators secured. These comprise prints of fingertips, fingers, and hands. Images courtesy of PHK Peter Eichardt, Police College of Saxony–Anhalt, Germany.

the interaction of "grabbing" alone would result in numerous ways to distribute and visualize fingerprints — to say nothing of distribution and visualization of blood traces.

Thus, a realistic presentation of trace data would require a simulation model, which would have to be connected to the player character's interactions. In the case of explicit authoring, an elaborate editor to specify trace data distribution and visualization would be required. In the case of implicit authoring, however, the different possibilities of trace transfer interactions would easily result in an excess of user input commands to control each possible interaction type. This would generally be difficult to implement and was not feasible in the scope of the *OpenCrimeScene* prototype. A simplified solution had to be implemented therefore.

Simplistic Approach First of all, trace data has to be restricted to a reasonable number of traces. In the case of *OpenCrimeScene*, these are

- fingerprints
- footprints
- blood traces.

A straightforward solution to present such data in the virtual environment is to use texture images as visual representation for each trace type and to map the texture images to the scene objects. To realistically present trace texture images, they would have to be blended with the scene object textures. This requires carefully pre-processing each scene object as described above. A simplistic solution is to map trace textures onto an intermediate geometric model, i.e., a 2D quad, and to add that model to the scene. For the time being, this avoids pre-processing of scene objects and difficult texture mapping

Figure 5.13.: Evidence textures used in *OpenCrimeScene*. From left to right blood trace, fingerprint, and footprints.

processes. However, the geometry of a 2D quad only matches plain surfaces and attracts attention when placed on a round surface.

In the course of her diploma thesis, Reinemann [2006] developed a basic trace concept that follows the previously described approach.[9] The author implemented a class hierarchy for fingerprints, footprints, and blood traces that are derived from a basic trace class to summarize common trace properties like latency and change over time. The different trace types are presented in the virtual environment as texture objects placed on a geometric quad. To simulate latency and changes over time, different transparency values can be specified by the user. Figure 5.13 illustrates the different trace textures in *OpenCrimeScene*.

Two further student assignments supervised within the scope of this dissertation dealt with the simulation model necessary to present trace data in the virtual crime scene realistically. These concerned examination and use of shader programming and texture mapping to authentically visualize fingerprints and implement the geometric distribution of blood traces. Footprint traces will be discussed in the context of implicit authoring in Section 5.3.3.

Realistic Presentation of Fingerprints

Generally, fingerprints are classified as

- latent prints caused by epidermic excretion substances (normally, this is a combination of sweat and skin particles) that are left on objects touched by humans
- visible prints caused by other liquid substances that are transferred from the hands of a human to an object when touched

9 The diploma thesis focused on user interaction and will be examined in detail in Section 5.3.3.

(a) (b) (c)

Figure 5.14.: The images illustrate the difficulty of recognizing fingerprints. Even on material like glass fingerprints often are hardly visible (a). Dusting the prints therefore is necessary (b) to clearly recognize and protect it (c). Images courtesy of PHK Peter Eichardt, Police College of Saxony–Anhalt, Germany.

- molded prints caused by impressions of hands left on malleable materials, e.g., dust, fat or grease, window putty, etc.

Especially fingerprints caused by liquid substance transfer or hand impressions are visible in most cases and could easily be visualized using colored texture images. In contrast, fingerprints caused by epidermic excretion substances are often latent. The prints' visibility depends on the surface material, viewing directions, and lighting conditions; typically, they can be recognized after dusting the surface material, as illustrated in Figure 5.14.

To realistically visualize latent fingerprints in a virtual environment, correlations of surface material, lighting conditions, and viewing directions have been examined in an experiment. According to the experiment, the fingerprint's visibility is mainly dependent on the angle between reflected light, viewing direction, and distance between viewer and fingerprint. Based on this a shader program was developed using the development tool *RenderMonkey*, which computes the angle between reflected light and viewing direction. The fingerprint texture only gets drawn if the angle is within a certain threshold range. Details of the experiment and shader program can be found in [Hewicker, 2007]. Resulting images are illustrated in Figures 5.15.

The shader program can be considered a first step into realistic fingerprint visualization that will serve as a basis for future developments. Especially, further parameters like the amount of epidermic substances and surface material need to be integrated. Also, bump mapping seems suitable to visualize molded prints. Such improvements will have to be included in *OpenCrimeScene* as part of future work in order to improve the educational effect of the system.

Figure 5.15.: Fingerprint latency depends on the material the print is on, but also on the incoming lighting direction, viewing direction, and distance between viewer and object. The images illustrate the distance aspect under different lighting conditions. The teapot is covered with fingerprints. Due to the different lighting direction that come from left, front, and right side, the fingerprints' visibilities change.

Realistic Presentation of Blood Traces

In addition to realistic fingerprint visualization, realistic geometric distribution of blood traces is equally important to support CSI training. Therefore, Brosch [2006] developed a general model to represent liquid drops and splashes in a virtual environment. The model includes several pre-defined parameters like velocity, consistency, and pressure to realistically calculate liquid flow. Direction of the flow is specified via picking: the ray between viewing center and picked coordinate serves as flow direction. According to the calculation, textured 2D quads are recursively added to the environment until a certain flow threshold is reached. Details can be found in [Brosch, 2006].

Like fingerprint visualization, geometric distribution of liquid is a first attempt and still requires improvement like explicitly defining the direction of liquid flow and splash distribution, for instance. In this regard, however, the previous sections show how challenging presentation and implementation of specific learning material data can get. Particularly, the scope of a prototype often calls for simplistic solutions to prove the developed concepts. In the case of *OpenCrimeScene*, for instance, trace data was required for implicit and explicit authoring of housebreaking, as will be described next.

Implicit Authoring

According to the extended game universe and resulting general system framework, implicit authoring focuses on the game playing aspect of authoring. The teacher has to be addressed as a game player who can pursue teaching and authoring as a game playing experience. In the case of *OpenCrimeScene*, implicit authoring has to make interactive housebreaking possible. To simulate realistic housebreaking situations, a central question that has to be discussed is, how to implicitly emulate trace transfer from player character to scene object as part of game playing such that the player is not necessarily aware of leaving traces at the crime scene? In reality, robbers are not necessarily aware of leaving traces at the crime scene.

Implementation of implicit authoring is challenging. As was already discussed in the preceding section, realistic implementation of trace transfer might result in an extensive combination of different user interaction types and user input commands to trigger such interactions. "Grabbing" alone poses a challenge to implicit authoring. First of all, the system had to provide different types of grabbing. Second, the player would have to explicitly select one type of grabbing in order to specify how to grab an object. Thus, even though the user might be immersed in the game story, unintentional behavior is difficult to produce and requires an elaborate user interface design. To approach these issues in the scope of the *OpenCrimeScene* prototype, implicit trace transfer has been restricted to a small number of interaction types and corresponding trace data.

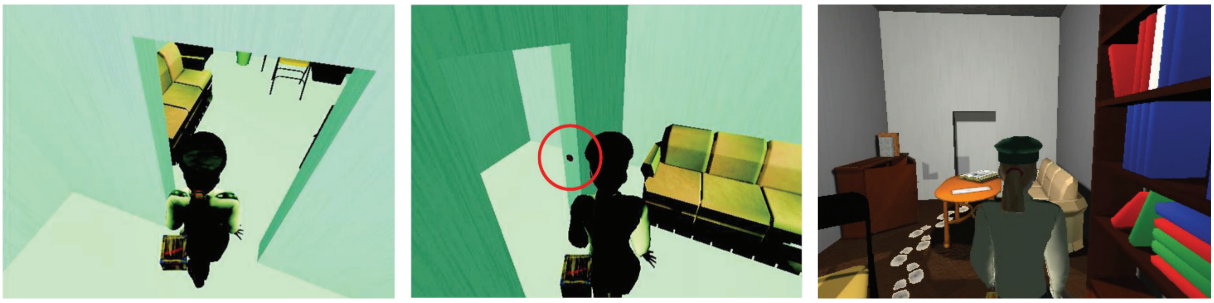
Implicit Trace Transfer In a virtual environment, implicit trace transfer is similar to adding a trace object to the scene as a result of a specified user interaction with the scene. Three implementation issues have to be considered therefore:

1. Defining user interactions that cause trace transfer.
2. Linking such user interactions to trace objects.
3. Adding trace objects to the scene whenever the user triggers a specified interaction.

In the course of her diploma thesis, Reinemann [2006] examined relationships between different types of traces and human behavior with respect to consciousness (cf. Reinemann [2006, pp. 39/40]). Three obvious types of human behavior that might cause trace transfer are grabbing, walking, and bumping. Consequently, in *OpenCrimeScene* three interaction types are specified to implicitly transfer trace data to the scene:

- selection of scene objects causes fingerprint transfer
- navigation through the environment causes footprint transfer
- collision between player character and scene object causes blood transfer.

Whenever object selection or character navigation is triggered by the user as input control, the system adds fingerprint or footprint texture images to the virtual crime scene. In the case of character-object-collision, blood traces are transferred as a proof of concept. Since *Delta3D* did not provide for automatic collision detection and response



(a) (b) (c)

Figure 5.16.: The images illustrate three possibilities of implicit trace transfer. The character carries an intermediate object to detect collisions between character and scenario (a). Whenever the character collides with a scene object, a blood trace is placed at the position of the collision (b). During walking the character leaves footprints on the floor (c). Images (a) and (b) courtesy of Reinemann [2006].

between player character and scene objects but only between scene objects, this type of interaction will have to be improved.

To overcome this problem, Reinemann assigns an intermediate scene object to the player character and manually detects collisions. Whenever a collision occurs, blood trace data is added at the respective collision position in the virtual scene. To add footprints to the scene, the current position of the character is used and footprint data is added to the floor whenever the user starts to navigate through the environment. Finally, fingerprint data is added to the scene whenever the user selects scene objects. Since picking is used to implement selection, the coordinate position to add the fingerprint is directly calculated. Figure 5.16 illustrates different trace data that were added to the crime scene.

In addition to trace transfer, the user can also remove and translate scene objects to simulate “stealing” and “destruction.” A so-called “robber’s bag” is available via the graphical user interface that contains the removed scene objects. This has been added to *OpenCrimeScene* to enhance housebreaking as game playing experience.

The introductory discussion on the challenges of implicit authoring in general and housebreaking in particular show that realistic reproduction of training situations in an interactive 3D environment provides numerous challenging research topics that primarily address user interface design. In this regard, it is equally conceivable though that implicit authoring cannot serve as a universal authoring technique for digital game-based training systems. Explicit authoring will be required in addition, to allow for specifically preparing training sessions. In the context of *OpenCrimeScene*, this primarily concerns explicit positioning of trace data, for example. The following section, thus, discusses the explicit authoring means required and implemented in the current training system prototype.

Explicit Authoring

In contrast to providing a game playing experience for the teacher, explicit authoring is intended to support the teacher in adapting training sessions to current lessons in a straightforward manner (cf. Sec. 3.2.2). The teacher has to be able to specify and determine training sessions and by this directly pursue teaching situations. In terms of *OpenCrimeScene*, explicit authoring has to address

- preparation of the virtual crime scene; primarily, that is, specification of trace data and positioning of trace data in a virtual environment
- preparation of assignments or tasks and quizzes, for example, using existing learning material.

Both aspects have to be provided for by an authoring interface such that the teacher can explicitly prepare virtual training sessions.

Virtual Crime Scene Preparation

Preparation of the virtual crime scene posed several challenges to implicit authoring. In the case of explicit authoring, especially, challenges of realistic trace transfer can be tackled by using an editing interface. Such an interface has to offer a set of trace data for selection as well as specification means to determine trace visualization and geometric distribution. Again, several interesting research topics for interface design result from this because the interface has to adapt to a specific educational situation and, additionally, has to address non-expert users.

For example, in the case of adding fingerprint data to the virtual scene, it was proposed to select a scene object and switch into editing mode to prepare the object. To realistically simulate fingerprint transfer to the scene object in editing mode, a *virtual hand* would be an ideal editing tool. Such a virtual hand would have to be controlled by the user to simulate “grabbing.” According to the user’s specification, the fingerprint data could then be transferred from the virtual hand to the scene object. Development and integration of further editing features are naturally open.

In the scope of the first *OpenCrimeScene* prototype, such elaborate explicit authoring means could not be implemented though. To nevertheless provide a first approach to

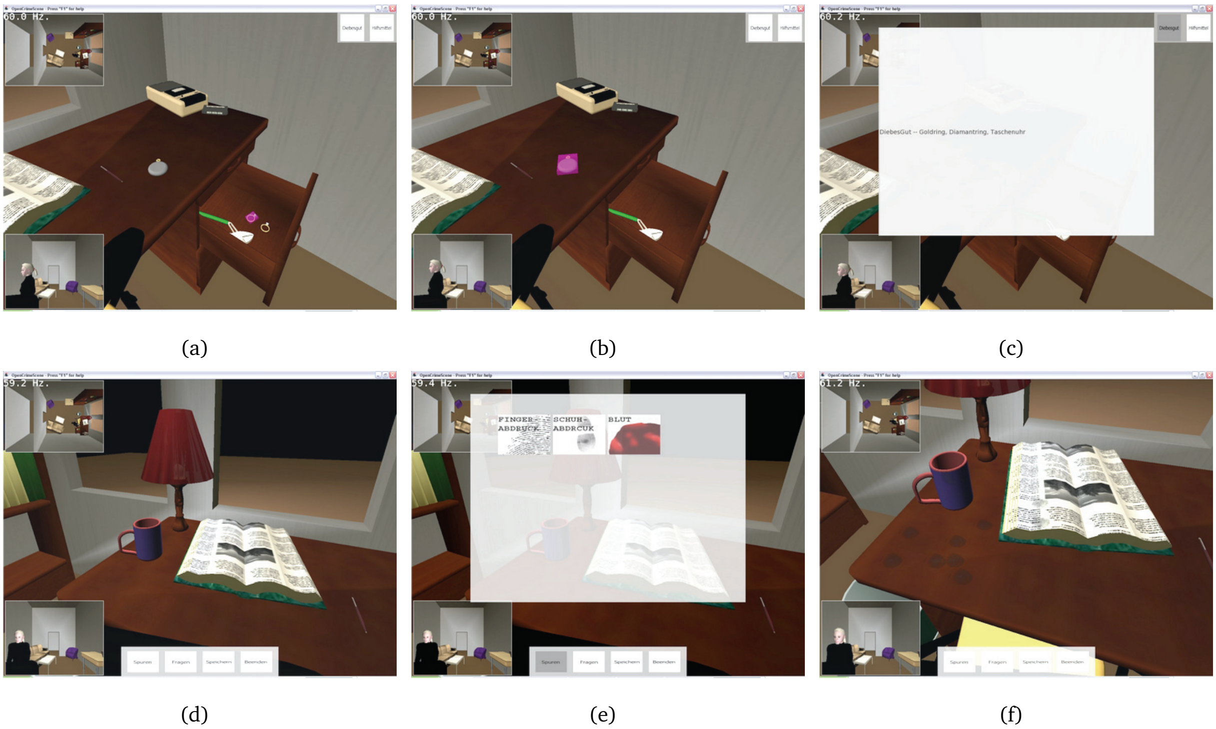


Figure 5.17.: The images illustrate explicit trace transfer. In the upper right corner the robbers “tool box” and “bag” can be accessed (a). When selecting an object in the scene, the object is highlighted and can be “stolen” (b). By selecting the robber’s bag, the robber can see what kind of objects have already been stolen (c). To explicitly leave traces at the scene, the teacher has to activate a bottom menu (d) which gives access to traces (e), quizzes, or task creation. After having selected a trace it can be added to the scene via picking, as illustrated by the fingerprint traces (f).

explicit authoring, the current prototype allows the teacher to explicitly add trace data to the scene. To do so, the teacher can select trace data from the back-story menu and add it to a scene object by selecting it via picking. Trace data then is added to the picked coordinate position. Although straightforward an approach, it demonstrates explicit authoring and serves as a basis for future improvements. The approach is illustrated in Figures 5.17.

Quiz Preparation

As stated above, explicit authoring also includes the preparation and insertion of quizzes for training sessions. In CSI training, transfer of legal issues to practical investigation is of primary interest because this is currently not provided by the syllabus of the Police College.

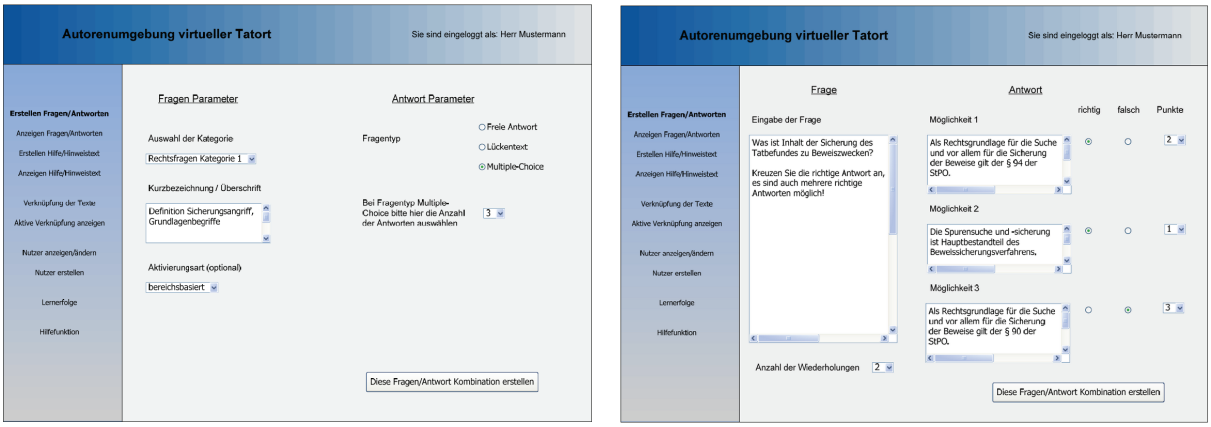
In general, the set of legal bases that establish crime scene investigation depends on the current situation of the investigation. Legal bases may change during investigation. For example, interviewing witnesses is based on a different legal basis than interviewing suspects. In the course of the interview, the interviewed person may change his or her role from witness to suspect. This means a change of the legal basis and might lead the investigator to switch from interview to interrogation. In *OpenCrimeScene*, such changes of legal basis have to be linked to the course of the game story. Development of an interactive quiz on legal issues, thus, means adapting the quiz to the current story situation.

In the course of his diploma thesis, Spengler [2006] developed a concept for authoring interactive quizzes in *OpenCrimeScene*. The thesis concentrated on intuitive authoring of legal questions and tried to allow for linking different types of questions according to possible changes in the legal basis. Spengler classifies three types of legal questions that are based upon different legal aspects:

- legal basis of crime scene protection
- legal basis of crime scene processing
- legal basis of individual examination

Based upon these categories, a quiz editor was developed. It allows teachers to select a category, pose a question, and assign one or multiple answers to it. Furthermore, each question can be given a certain priority. Due to the specified priority, and given that further questions in that category exist, the system proposes subsequent questions. This facilitates preparation for changes in the legal basis, although it does not provide for automatically linking questions to game story progress. The interactive quiz was prototypically implemented using PHP as user interface and SQL as database interface. An integration of the quiz into *OpenCrimeScene* is still due though. Therefore, questions need to be added to certain game events caused by user interaction. Details of the concept are given in [Spengler, 2006]. Figure 5.18 illustrates the PHP-based graphical user interface for authoring questions.

The foregoing sections demonstrated that in-game authoring requires long-range planning and extensive implementation effort. In doing so, it poses challenging research questions to user interface design and realistic presentation of data. The prototypical implementation of *OpenCrimeScene* touches aspects of in-game authoring. The prototype nevertheless proves that in-game authoring can be considered a novel approach to content authoring. Especially, the combination of implicit authoring and explicit authoring techniques facilitates content authoring and addresses non-experts as well as advanced users.



(a) (b)

Figure 5.18.: The images illustrate the graphical user interface of the quiz tool. The teacher can select certain categories of questions (a) and pose questions as well as correct answers or additional help texts (b). Images courtesy of Spengler [2006].

In the context of digital game-based learning, in-game authoring gives teachers a helping hand to use a digital game-based training system. Familiarity with these software applications can be built up through game playing experience, yet without neglecting the educational purpose. In-game authoring most likely might also be an adequate solution to enhance other authoring situations, for example, level editing or story design like in role playing games (cf. Sec. 4.3). Unlike in-game authoring, in-game training pursues a straightforward approach as will be discussed next.

5.3.4 Training Component

Unlike the authoring component, the training component has not seen an explicit separation between game playing and training experience up until now. As was discussed in Section 4.4, it should motivate students to continue digital game-based training and let students consider the digital game-based training system as an educational tool. To do so, game playing elements have to be interwoven into training experience and, thus, into explicit links to education. In that respect, the main-story has to be considered as the primary motivational aspect: the student plays against the teacher and has to solve the problems posed. In the case of *OpenCrimeScene*, the student has to solve the crime. Authentic data presentation seems to be essential to support the student in the digital game-based training system as an educational tool.

The challenges of realistically presenting human behavior and trace transfer were discussed particularly in the context of the authoring component. Such challenges hold true for the training component because police students have to learn not to contaminate the crime scene with their own traces. However, in regard to time constraints, the focal point of the training component was put on the realistic implementation of forensic tools and criminalistic procedures in the scope of the first prototype. As a result, the training component addresses two training aspects:

Identifying evidence: Prior to processing any kind of evidence, investigators have to identify evidence. A common technique is use of additional light, for example, using a torch to illuminate the scene. Especially footprint and fingerprint evidence can be identified more easily then. Generally, material modifications caused by evidence traces result in different kinds of reflections that can be noticed with additional lighting.

Documenting evidence: To document evidence, investigators traditionally take pictures of the crime scene and of evidence found at the crime scene to better being able to reconstruct the crime. Crime scene photography, thus, is central part of police education. It teaches police students how to handle SLR cameras and how to approach the crime scene photographically.

Criminalistic procedures that concern these two aspects have to be integrated to *OpenCrimeScene*. Obviously, a third aspect central to CSI education is processing evidence. Processing evidence concerns, for example, dusting and taping fingerprints or collecting blood particles. Since these procedures are closely related to realistic trace data presentation, they will be tackled as part of future work. As a first step into that direction, forensic tools have to be modeled and integrated into *OpenCrimeScene*.

Forensic Tools

Forensic tools are essential when it comes to collecting and processing evidence. These tools have unique features regarding their appearance and functionality, and police students have to acquire detailed knowledge about how to apply specific tools to specific evidence traces. In regard to interactive use of such tools in *OpenCrimeScene*, authentic 3D models of each tool are necessary. In her internship report, Grzeschik [2006] examines each of the forensic tools as well as their different application areas and describes the modeling process in detail. The Police College supplied a set of photographs of the

Figure 5.19.: A subset of forensic tool models (below) compared to real tools (above). The images illustrate chalk, gas, magnifier, and spoon. Upper sequence of images courtesy of PHK Peter Eichardt, Police College of Saxony–Anhalt, Germany. Lower sequence of images courtesy of Grzeschik [2006].

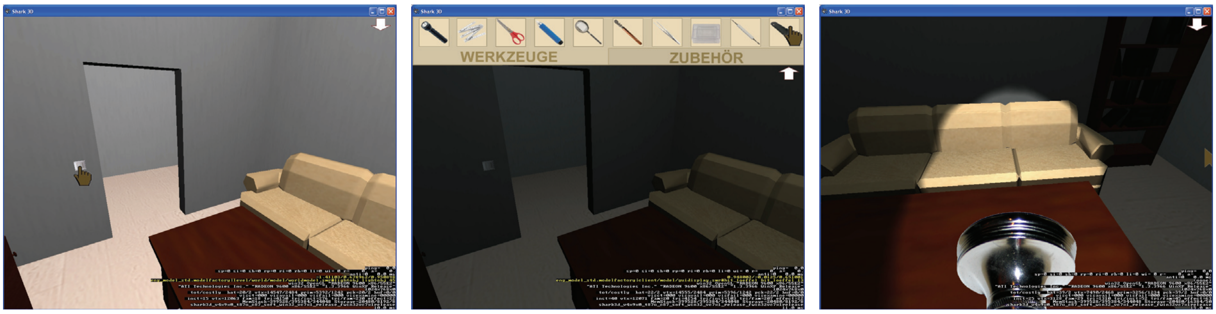
regular tool set, which were used as texture images. Figure 5.19 illustrates the set of forensic tools developed for *OpenCrimeScene*.

To actually allow for evidence processing, each tool needs to be assigned a processing functionality like, for example, dusting. Possible implementation approaches will be discussed in Section 6.2. So far, the torch tool can be used to identify evidence, as will be described in the subsequent section.

Virtual Torch

The virtual torch has to serve as an interactive spotlight that can be controlled by the cop character. The implementation of a virtual torch-light was conducted at an early development stage of *OpenCrimeScene* that used the *Shark 3D* engine as a development tool. Also, no character had yet been added to the prototype. Therefore, the torch is represented as a texture image at the bottom of the screen when the user activates it. While navigating through the environment, the torch light can be used to illuminate the scene. Implementation details can be found in [Grzeschik, 2006]. Selection and use of the torch as an interactive tool to light the scene is illustrated in Figures 5.20.

So far, the virtual torch exemplifies how forensic tools could be used in *OpenCrime-Scene*. Integration of the virtual torch into the current *Delta3D*-based prototype is still pending and will be part of future work. Especially interactive use of the torch has to be implemented, which requires adding appropriate animation to the character like ducking or swinging the torch, for instance. Apart from this tool, a virtual SLR camera has been developed that realistically simulates a real SLR camera and allows students to produce authentic photographs.



(a) (b) (c)

Figure 5.20.: The image sequence illustrates use of the virtual torch. The user has to turn out the light in the virtual scene to enhance the effect (a). Via the graphical user interface, the torch tool can be selected (b) and is displayed at the bottom of the screen (c). Images courtesy of Grzeschik [2006].

Virtual SLR-Camera

As was already mentioned, crime scene photography is an individual lesson in police education that teaches police students how to document the crime scene photographically. In order to integrate this aspect of police education into *OpenCrimeScene*, a virtual SLR (single-lens-reflection) camera has to be developed that simulates a realistic camera model. It has to allow police students to specify camera parameters and take pictures of the virtual scene. The basic requirement of a virtual SLR camera is real-time simulation of lens effects, which can be achieved using GPU-based shader programming. Development and implementation details have been published in [Brennecke et al., 2008a]; further details are given in [Panzer, 2007]. The author made a major contribution to the so-called vSLRcam.

Central to vSLRcam is the simulation of camera components and their interrelations. Camera components that are essential to creating certain lens effects are *camera lens*, *aperture*, *shutter*, and *film type*. By specifying the parameters of such components, photographic effects can be produced such as *angle of view*, *exposure*, *motion blur*, and *depth of field*. In order to serve as a training tool, vSLRcam has to realistically produce these effects based on user-specified parameter settings. The following list of components and according camera parameters is integrated into vSLRcam and can be parameterized by the user like in reality:

- camera lens that is determined by focal length f and distance between lens and focus plane d_s
- aperture (size) that is determined by f-number a

(a) Wide angle lens (b) Normal lens (c) Tele lens

Figure 5.21.: Simulation of different camera lens types resulting in different angles of view. The camera position is the same for each picture. Images courtesy of Panzer [2007].

- shutter speed that is determined by exposure time t
- film type that is determined by ISO value i_s and film format i_f.

Further parameters have to be derived to simulate lens effects. Yet the following paragraphs mainly present the results of the method; for further details refer to [Brennecke et al., 2008a].

Angle of View The angle of view is given by the camera lens type and is responsible for the viewing pane of the resulting image. Adjustment of the angle of view can be implemented by changing OpenGL's view frustum accordingly. Figure 5.21 shows different lens type simulations.

Depth of Field In contrast, rendering depth of field is more complex. Depth of field describes the area surrounding the object *in focus* that is sharp in the final image. Objects that do not fall into that area will become blurry in the final image. This is caused by lens projection: each object point that gets projected onto the image plane forms a so-called *circle of confusion*. Objects that are in focus cause small circles of confusion and, thus, sharp areas; objects that lie out of focus cause large circles of confusion and, thus, blurred areas.

Consequently, simulation of depth of field requires calculating a circles of confusion for each image point. Therefore, further parameters have to be derived based on the user-specified parameter setting of lens type, aperture setting, and lens-to-focus-plane distance. Based on these parameters, a circle of confusion is calculated for each image point. To actually blur the image, a Poisson disc filter is applied to each circle of confusion in a second step. Results are illustrated in Figure 5.22.

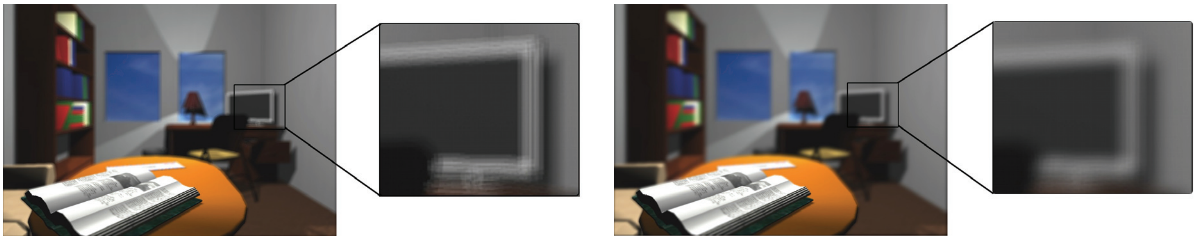
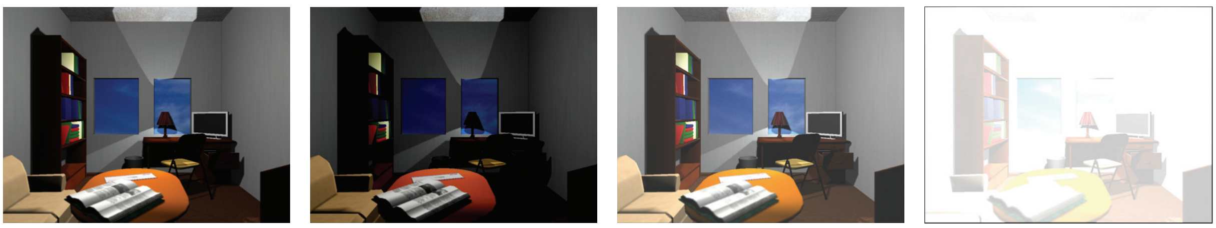


Figure 5.22.: Both pictures contain depth of field. Due to the Poisson disc filter the left picture consists of regular artifacts. By re-sampling the image, the artifacts can be suppressed (right image). Images courtesy of Panzer [2007].

Exposure The next effect that has to be integrated into the vSLRcam is a realistic simulation of the exposure. In reality, during exposure the image plane of a camera is exposed to the incoming light. If the amount of light is too high, the final image becomes over-exposed. If it is too low it becomes under-exposed. To achieve correct illumination, the photographer has to carefully adjust aperture size to specify the amount of incoming light, and shutter speed to specify the exposure time.

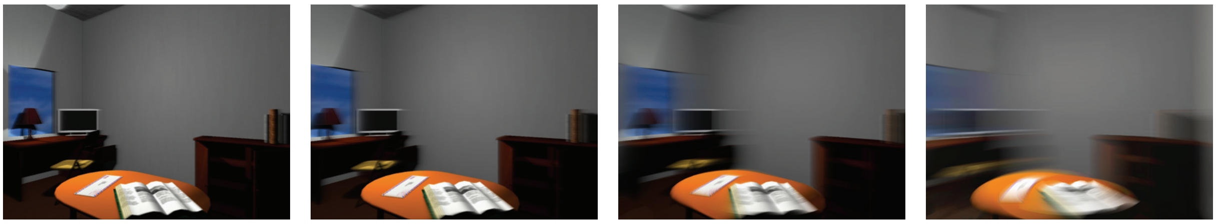
To simulate exposure, two exposure values have to be calculated, which are determined by the general scene illuminance and the user-defined combination of aperture and shutter speed. A balanced exposure is given if the difference of both values is minimal. Calculations further have to include the film type and a photometer dependent constant. Since a correct specification of the scene illuminance would require a global illumination model and since the underlying OpenGL API is based on a local illumination model, the exposure value for scene illuminance is an approximation. The results do however come close to realistic photographic exposure, as illustrated in Figure 5.23.

Motion Blur Finally, motion blur is the last and most difficult effect that has to be added to vSLRcam. It typically occurs during exposure when either scene objects or



(a) correct exposure (b) under-exposure (c) weak over-exposure (d) heavy over-exposure

Figure 5.23.: The illustration shows four types of exposure. Figure (a) shows the first rendering pass image. This becomes under-exposed with settings $a = 4,0f$ and $t = \frac{1}{60}$ (b). Moreover, (c) and (d) are over-exposed by values $a = 2,0f$ and $\frac{1}{60}$ and $a = 1,4f$ and $t = \frac{1}{15}$. Images courtesy of Panzer [2007].



(a) $t = \frac{1}{30}s$ (b) $t = \frac{1}{10}s$ (c) $t = \frac{1}{5}s$ (d) $t = 1s$

Figure 5.24.: Simulation of motion blur by varying exposure times t. Images courtesy of Panzer [2007].

the camera move. Each change of position of scene objects is captured on the image plane and results in blurred areas. This is due to the fact that during exposure the individual illuminance values are added up over (exposure) time. To simulate this effect, individual illuminance values of each frame that is rendered during exposure time has to be accumulated in a final image. Again, scene illuminance values can only be approximated. To do so, implementation faces two challenges:

1. The contribution of each frame to the final image has to be decreased by the number of rendered frames.
2. The decrease of each frame's contribution to the final image still has to ensure that neither frame becomes transparent.

To decrease each frame by the number of rendered frames, a weighting factor $\alpha : [0, 1] \rightarrow \mathbb{R}$ is introduced. This factor is due to two conditions:

1. Given the exposure time, the frames per second determine the number of frames that have to be accumulated into the final image. For example, an exposure time t of $\frac{1}{8}\,s$ and 80 frames per second results in $n = 10$ frames that have to be accumulated.
2. The color intensity of an image is nearly transparent if reduced to $\epsilon = \frac{1}{50}$ of the original value and becomes invisible if reduced further. Hence, the first frame that is weighted by α should not fall below $\frac{1}{50}$ of its original value.

Applying these conditions, the weighting factor of the first frame has to be specified using $\alpha = \sqrt[n]{\epsilon}$ to assure its contribution to the final image (cf. Brennecke et al. [2008a] for a detailed explanation). Implementation of the motion blur effect could then be done in real-time; resulting images are presented in Figure 5.24.

Panzer [2007] implemented the vSLRcam as a standalone-application that was integrated into *OpenCrimeScene* as special forensic tool. Pictures taken with the vSLRcam have

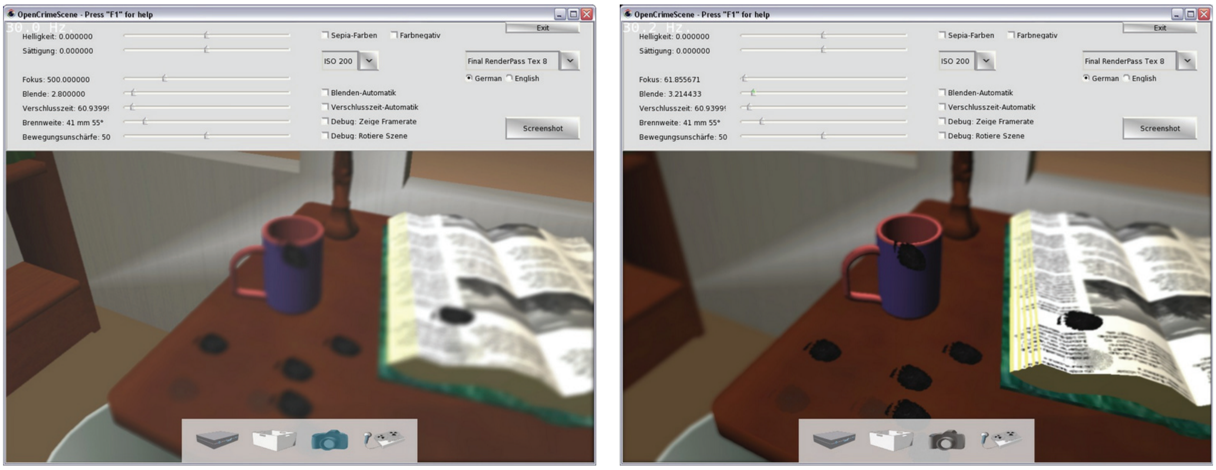
(a) $a = 3,5\,f, t = \frac{1}{200}$ (b) $a = 3,5\,f, t = \frac{1}{60}$ (c) $a = 3,5\,f, t = \frac{1}{30}$ (d) $a = 3,5\,f, t = \frac{1}{4}$

Figure 5.25.: Comparison of the vSLRcam with a real SLR camera. The upper images are taken with a Nikon D70, whereas the lower images are virtual shots. As can be seen, the scene illuminances correspond very well. Images (a) and (b) show under-exposure and correct illumination, whereas images (c) and (d) show slight and strong over-exposure. The camera settings were the same for the real shots as well as for the virtual ones. Images courtesy of Panzer [2007].

been compared to real photographs taken with a digital SLR camera, the Nikon D70, to exemplify the realistic simulation of photographs (see Fig. 5.25). Camera parameters have been set identically and lead to very similar images. Further results are provided in [Panzer, 2007].

As such, the vSLRcam can serve as an adequate training tool to support crime scene photography training. Students can by themselves engage with the vSLRcam to understand how different parameter settings correlate. In addition, if access to real SLR cameras is restricted to lessons only, students can use the vSLRcam for training purposes. Figure 5.26 illustrates the GUI of the virtual SLR camera, which allows for different parameter specifications and taking snapshots.

The preceding sections describe in-game training as a straightforward digital game-based learning approach that offers realistic training material in a game environment to reinforce and enhance training processes. Yet unlike common tools used in digital game-based learning like, for example, serious games, in-game training as proposed by the extended game universe tightly links the training processes to education. This is primarily achieved by introducing back-story and main-story and letting the student play against the teacher. Influences on the motivational effect are conceivable, but will have to be evaluated in extensive user study.



(a) (b)

Figure 5.26.: The image illustrates the current graphical user interface of the virtual SLR camera. Among others, the user can produce blurry images by either zooming or changing the aperture size (a). After specifying the parameters, the objects become focused and the picture can be taken (b).

The implementation of the training component faces similar challenges as does the authoring component regarding user interface design and authentic data presentation. It also requires long-range planning to fit educational situations. This, however, is even more true for the third major component of the general framework: the reviewing component.

5.3.5 Reviewing Component

If the authoring component addresses the teacher as the main user and the training component addresses the student as the main user, the reviewing component is intended to forge links and to address both users, as proposed by the extended game universe (cf. Sec. 3.1.4). To do so, the reviewing component has to generate a visual review log of the training session that can be used for self-evaluation or classroom discussion and directly relates virtual training to real education. However, as was discussed in Section 4.5, the development of assessment and visualization strategies is challenging because several different research topics have to be addressed like data mining and information visualization. In addition, reviewing is based on input from authoring and training components.

As the previous sections have shown, the development and implementation effort to establish even basic solutions for authoring and training components is significant. As a consequence, in the context of *OpenCrimeScene* only partial solutions to review training scenarios have been developed so far. In particular, a general concept for an *OpenCrimeScene* review log has been developed and was published in [Brennecke et al., 2007]. Furthermore, an approach to simplify NPR line drawings has been developed that could be used to integrate visual highlights into the training system as well as to support rendering of the visual review log in general. Details have been published in [Isenberg and Brennecke, 2006; Isenberg et al., 2005].

In the course of developing the reviewing component for *OpenCrimeScene*, two further diploma theses have been supervised by the author of this dissertation. These concerned a general examination of training session assessment in serious games as well as the development of a basic task representation including a basic reviewing system. Details are given in [Sickel, 2006]. Moreover, different visualization forms for the visual review log to support teacher-student-communication have been developed. Details are given in [Müller, 2006]. The subsequent sections present reviewing solutions developed for *OpenCrimeScene*.

Logging, Replaying, and Representing Tasks

Reviewing comprises data specification, data evaluation, and data visualization. Reviewing functionality has to be examined, to estimate implementation costs of each aspect. In the course of his diploma thesis, Sickel [2006] developed a basic reviewing system that allows logging and replaying user interaction events in a virtual environment. Prior to logging, data have to be specified, such as mouse and keyboard device input, player positions, and camera coordinates. The system then logs these data in user-specified time intervals and stores them in an external log-file. The log-file serves as input for data assessment and data visualization.

A first step to assess training session data is to define task representation and to assess task accomplishment. Therefore, Section 4.5 suggested use of task events. Sickel [2006] conceptualized a task representation for evidence collection in *OpenCrimeScene*. The representation suggests establishing an ordered list of forensic tools, evidence objects, and additional time constraint. To accomplish a task, the user has to select the specified tools in the correct order and apply them to the evidence objects in a specific amount of time. To assess whether a task has been accomplished or not, task events have to be defined and linked to each forensic tool and evidence object. One task has to be set up

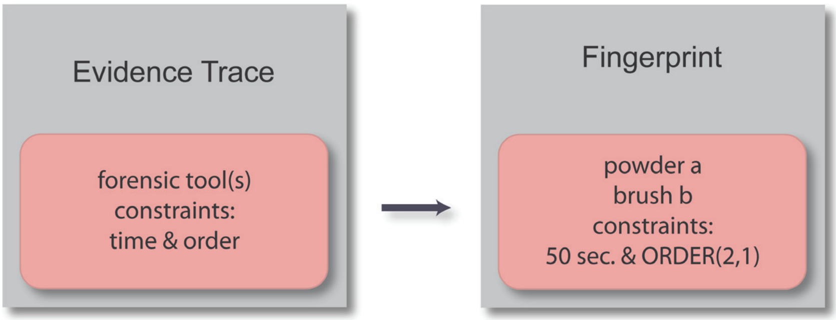


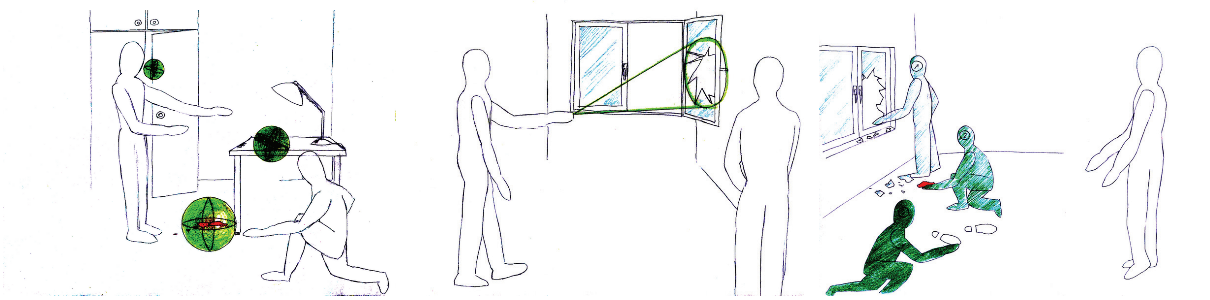
Figure 5.27.: The illustration shows the basic idea of representing tasks in *OpenCrimeScene* by Sickel [2006]. Constraints like order of task events and time are proposed to determine investigation tasks.

as a map or list of ordered task events. Consequently, the logging mechanism has to log each triggered task event and the current step in time. Figures 5.27 illustrate the concept. So far, the system allows the replay of the whole training session without assessing data explicitly. This is due to the additional effort that would have been required to implement data assessment, as discussed in [Sickel, 2006].

In addition to such technical challenges, virtual CSI training poses another challenge to reviewing techniques. Crime scene investigation is determined by a set of criminalistic procedures. Some of these procedures follow a strict order like evidence collection or taking pictures of the crime scene. However, tasks like conducting a first walk-through or observing the crime scene can hardly be specified in a strict order because the order of execution depends on the individual crime itself. These tasks require summarizing the overall user behavior. An assessment strategy will have to address course-based user interaction. Although not tackled in regard to user interaction assessment, different visualization forms have been developed that address course-based user interaction, among others.

Visualization Forms

Reviewing has been introduced as technical correspondence to teacher-student-communication. To explicitly support this kind of communication, user interaction like grabbing or dusting fingerprints have to be visually emphasized. To do so, visual elements have to be integrated either into the training system to support synchronous communication, or into the visual review log to support asynchronous communication.



(a) (b) (c)

Figure 5.28.: Communication enhancement metaphors: trace-mark metaphor (a), sight-mark metaphor (b), and protocol-mark metaphor (c). Images courtesy of Müller [2006].

Visual Communication Support

Visual communication support first of all requires conceptualizing different visualization metaphors. In the course of her diploma thesis, Müller [2006] conceptualized three different enhancement metaphors, therefore:

Trace-mark metaphor enhances user interaction events in the virtual crime scene. It is conceptualized as an extra 3D object, which surrounds the current user interaction (see Fig. 5.28(a)).

Sight-mark metaphor highlights the current field of sight of the user. It is conceptualized as a 3D cone-like pointer with variable size to put emphasis on certain aspects of the scene and follows the pointer techniques presented in [Olwal and Feiner, 2003; Smith et al., 2004] (see Fig. 5.28(b)).

Protocol-mark metaphor supports asynchronous communication. It explicitly addresses teacher-student-communication. It combines different user interaction events in an overview image (see Fig. 5.28(c)).

The first and second metaphor address integration of explicit visual communication elements into the training system. This would be especially useful to support a multi-player digital game-based training system or examination situations where the teacher has to check what the student is doing online. In this regard, both metaphors address future approaches to visualize reviewing. On the other hand, since realistic user interaction is difficult to simulate, such visual elements could be used as a description tool by which the student can leave a note on what interaction was intended. This idea will also have to be pursued as part of future work.

The protocol-metaphor has been applied to *OpenCrimeScene* to visualize course-based user interaction in an overview image. Therefore, different character positions have to be combined in one image, using a static camera position to record character movement. Time has to be encoded to make an assumption about the overall training process. Quite naturally, the character's rendering has to be adapted according to the different time steps in order to convey the character's behavior over time and to convey that the image does not depict multiple characters. To encode time different visual styles can be used such as:

Transparency: The character is depicted as highly transparent in the early stages of the training session and becomes opaque as time passes. The visualization is controlled by a linear function that maps time to the level of transparency. The function needs to be designed in such a way that the transparency value of the first position is not 100% in order to ensure the visibility of all positions.

Color: A color scale can be used to encode time by mapping color values to points in time. Given the chronology of character positions, the color values reflect the respective character positions. Therefore, the character has to be desaturated to achieve a gray value version and then be re-colored in the respective color.

Silhouettes: One major concern when superimposing multiple positions of the character in the same image is that the character might not be optimally visible in a rather cluttered image. To make the character stand out against the background, silhouettes can be added to enhance its visibility. The visual design of these silhouettes could, for example, be set to match the color of the character or could be used to map different values like importance or duration of time.

The rendering styles presented show different approaches to encoding time. Transparency seems to be the main indicator for time-sensitivity. Yet transparency is limited when it comes to visualizing events which are very close in time. Silhouettes can additionally be applied either to better differentiate between close positions or to directly encode time-lags. However, the latter does not hold true for short changes in time. Figure 5.29 shows a combination of both techniques with an additional legend relating transparency and silhouette colors to the actual time values. This seems to be an adequate visualization at least for a small number of significant events. To add further visual elements, line stylization using G-strokes could serve as solution for future visualization forms. Details of the visualization concept are given in [Brennecke et al., 2007].

Line Stylization Using G-Strokes

In common NPR line rendering systems geometric properties of 3D input data is used to extract and stylize edges. The stylization

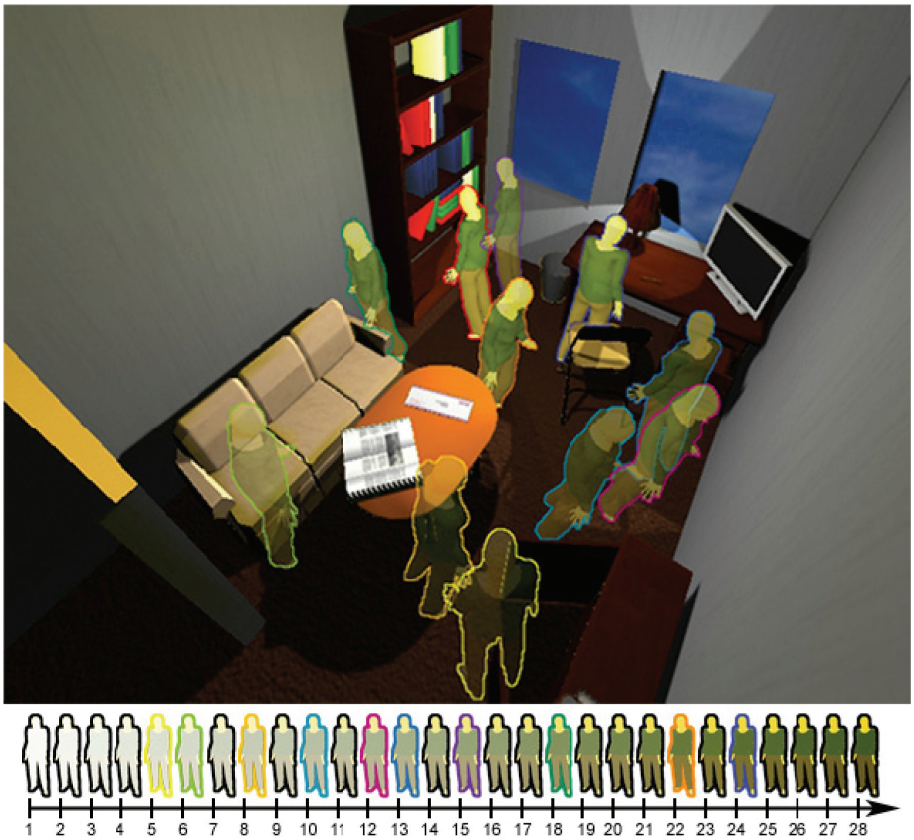


Figure 5.29.: The image illustrates the *OpenCrimeScene* review log. Different character positions are combined in one image. Transparency is used to differentiate time values; additional silhouettes emphasize the different character positions. The color of the silhouettes is also mapped to time values, as illustrated in the accompanying time scale. Image courtesy of Müller [2006].

of edges is generally restricted to the currently extracted edge though. This means, whenever different styles are to appear concurrently in the same rendition, certain computations have to repeated numerous times. This is not reasonable for 3D virtual environments and, consequently, the generation of renditions is often constrained to one or two styles to keep computational cost low.

To simplify line stylization from 3D input data, the concept of *G-strokes* has been developed [Isenberg and Brennecke, 2006]. G-strokes broaden the possibilities of generating expressive line drawings and keep computational cost low. All edges are kept and the geometric properties are extracted instead. Line stylization is carried out according to the extracted geometric properties, which means one style can be applied to a particular set of edges and another style can be applied to another set of edges without having to extract the designated edges anew.

In the context of enhancing user interactions in the virtual environment, G-strokes can be used to render stylized line drawings of selected scene objects, for example. Such stylizations can then be integrated into the visual review log to steer the user's attention or simply to emphasize specific user interactions. In addition, line stylization could be used during training to steer the traineés attention or to encode hints in the virtual environment to support training strategies.

Like the authoring and training components, the reviewing component poses several challenges to the development of a digital game-based training system. In the scope of this dissertation only a subset of such challenges could be faced. Especially the visual review log shows that visual assessment of training session could be effectively used for classroom discussion. This was also evaluated in a first usability study, which will be described subsequently. Before going into detail on the study, a brief system overview illustrates the current state of *OpenCrimeScene*.

5.3.6 System Overview

This section briefly outlines the *OpenCrimeScene* system state. In relation to the illustration of the general framework in Figure 3.3 (cf. Sec. 3.2), the approaches developed and implemented to support CSI training with *OpenCrimeScene* are illustrated in Figure 5.30. System functionality is presented as demo video as well as executable on the accompanying CD-ROM of this dissertation.

OpenCrimeScene is based on the general framework for digital game-based training systems, as was discussed throughout the previous sections. As a result, the prototype is a modular system that can be easily extended. The illustration depicts all contributions made to the *OpenCrimeScene* prototype. As can be seen, numerous elements have been developed that fit into the general framework's structure, especially, regarding the authoring, training, and reviewing components. The contributions range from basic modeling tasks to GUI design to several implementations that target the specifics of CSI training.

The current state of the prototype exemplifies the applicability of the general framework and additionally introduces several new approaches to data presentation and human-computer-interaction in the context of CSI education, even though not all concepts could be fully implemented and integrated into the prototype. The illustration clearly depicts that even a limited scope of system features makes considerable demands on the

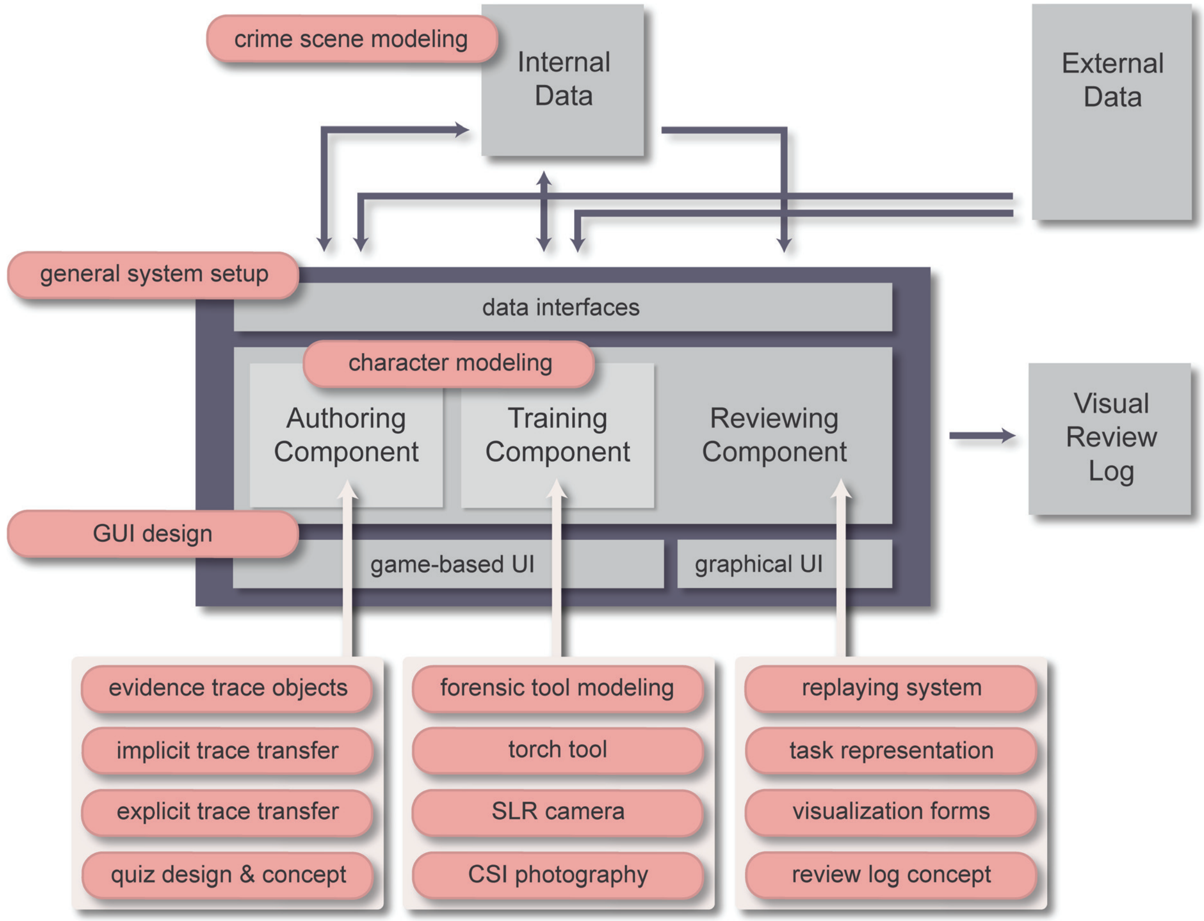


Figure 5.30.: The illustration depicts the *OpenCrimeScene* system based on the general framework. Basic data modeling has been conducted for internal data. The general system setup allows for accessing and storing back game level data. GUI design as well as game playing functionality has been added to the authoring and training component, respectively. In addition, functionality required for reviewing training sessions has been developed.

implementation. As was mentioned at the beginning of this chapter, this could not have been achieved without several contributions made by students as part of their student assignments.[10] Yet due to the modular design of *OpenCrimeScene*, further developments can be added easily and straightforwardly to each of the components. Hence, the current system state can be regarded as a basis for future implementations that contains essential and primary functionality to support CSI education.

Based on the *OpenCrimeScene* prototype's executable and demo video, a system evaluation was conducted to assess the prototype as well as the general framework for digital game-based training systems. The next section goes into detail on that.

10 In this context, I would also like to mention Sebastian Hundt who has partially worked on *OpenCrime-Scene* as student assistant.

5.4 Conducting System Evaluation

As proposed by the workflow schedule, the development of a digital game-based training system always has to be accompanied by regular system evaluations (cf. Sec. 4.7). Different methods exist to evaluate a software application. Software *usability* is evaluated, if the software targets a certain user group and, thus, is of primary interest for digital game-based training system development. Usability evaluation methods are used to ensure that users approve of the software design and, in that sense, are almost always related to evaluating the user interface.

The following subsections present general considerations on choosing a suitable usability evaluation method and go into detail on the evaluation of *OpenCrimeScene* as well as on the evaluation results.

5.4.1 General Considerations

Typically, two approaches are pursued to evaluate software usability: *usability inspection* and *usability testing* [Hegner, 2003; Nielsen, 1993]. Usability inspection addresses early development stages and does not require an executable prototype or full-functioning software application. A set of a few expert users inspects the software application and judges it, for example, according to certain usability principles, so-called *heuristics*. Results of such an *heuristic evaluation* can then influence software development at early stages and at low cost [Kosara et al., 2003; Zazelenchuk, 2006].

Usability testing, on the contrary, addresses later development stages and requires an executable prototype or full-functioning software application. It is an empirical method to evaluate real software use and requires an adequate number of end-users. Usability testing is more time-consuming in regard to preparation, conduction, and analysis of the evaluation, which is also more expensive than usability inspection. However, it is considered to be more accurate, too [Hegner, 2003].

The *OpenCrimeScene* prototype can be considered to be at an early development state: basic functionality is already provided by the system; additional concepts are queuing up for integration. This makes the prototype suitable for usability inspection and, in particular, for the inspection method of *heuristic evaluation*.

Heuristic Evaluation

Heuristic evaluation was developed by Nielsen and Molich [1990] and later refined by Nielsen [1994]. The approach is based on ten heuristics upon which a given interface or software design can be rated. The interface can be presented as paper or software prototype, demo video, or software application [Nielsen, nd]. Nielsen [1993] recommends up to five experts who have to inspect the user interface in place of end users. The author confirms that further experts would neither find more usability problems nor change the overall results of the inspection in any significant way. According to Squires and Preece [1999], heuristic evaluation leads to good predictions of end user problems by addressing key usability issues (cf. also Nielsen and Mack [1994]). As such, it is regularly referred to as the standard inspection method in literature [de Villiers, 2007; Hegner, 2003; Pinelle et al., 2008; Zazelenchuk, 2006].

The standard heuristics by Nielsen [1994] have been adopted to educational software evaluation as well as to game design by several researchers [Albion, 1999; Ardito et al., 2006; Desurvire et al., 2004; Pinelle et al., 2008; Silius and Tervakari, 2003; Squires and Preece, 1999; Ssemugabi and de Villiers, 2007]. Quinn [1996] established eight heuristics to evaluate *educational software design*. Based on this, Albion [1999] developed nine heuristics to further evaluate the *educational content* of such software applications. The author proposes judging educational software based on user interface design heuristics, educational software design heuristics, and educational content heuristics. The different sets of heuristics can be found in Appendix B.

Evaluation of OpenCrimeScene

The evaluation of *OpenCrimeScene* is the first evaluation of the prototype, which is intended to assess the prototype by collecting expert feedback. Throughout the development of *OpenCrimeScene* regular meetings with the project leaders both from the Police College and from the University of Magdeburg were held to ensure that the prototype suited user requirements. Yet, to make a general assumption on the system's design, this limited circle of experts has to be expanded to include

- police teachers
- police students

- developers.

The expert groups of police teachers and students are obvious choices because they will be the end users of the prototype and have to rate it according to what suits them best. Therefore, the focal point of the questionnaire will have to be on addressing these two expert groups. The expert group of developers has been added to ensure that the user interface design meets general requirements, which might have been overseen during the implementation. Also, this group is assumed to be well acquainted with novel system design standards and, thus, is expected to give additional and valuable information on technical aspects. This seems to be of special interest in the technical context of the general framework, which represents a novel approach to digital game-based training system design.

Since no previous evaluation results can serve as a basis for the inspection, use of well-known heuristics seems to be a good starting point to assess the system. However, assumptions that were made when establishing the user working profile also have to be evaluated (cf. Sec. 5.1). Thus, the evaluation has to address different aspects and is not intended to focus on Nielsen's heuristics. Moreover, since the main goal of the evaluation is to collect expert feedback, a questionnaire is assumed to be an adequate solution. The design of the questionnaire as well as the evaluation goals pursued will be considered in detail in the following section.

5.4.2 Design of the Questionnaire

As was mentioned, the questionnaire has to survey different aspects to evaluate *Open-CrimeScene*. Therefore, evaluation goals have to be settled first of all. Since this dissertation is concerned with the development of a general system framework for digital game-based training system that is based on the extended game universe concept, it has to be ensured that the assumptions made about the software knowledge of teachers and students, as well as the general willingness to use digital game-based training systems in class, are accurate. Moreover, the interface design of the prototype has to be evaluated as well as its usefulness and applicability to education. In that regard, the novel concepts developed in this dissertation also have to be assessed. As a result, the questionnaire is subdivided into four categories with each category targeting an evaluation goal. The complete questionnaire is included in Appendix C and will also be part of the discussion on the results in the next section. The categories and evaluation goals split into:

General questions: These questions are used to evaluate the general assumptions made during the establishment of the user working profile. This category first asks for general information like age, gender, and membership of expert group. This is followed by six questions that have to be rated. These concern the general and particular knowledge of PC software and computer games as well as whether PC software or computer games have been used in class so far. Also, experts have to give an estimate on the effect of using such applications in class.

Usability principles: These questions are used to evaluate user interface and application design. This category comprises ten questions that are based on the ten heuristics developed by Nielsen [1994] (cf. App. B). Experts have to rate whether the user was always aware of what is going on ("visibility of system status"), whether the user understood menu information presented by the system ("match between system an real world"), or whether the application's graphical design appealed to the user ("aesthetic and minimalist design"), for example.

Usefulness for education: These questions are used to evaluate the applicability and usefulness of the system to education. The category includes six questions, which adopt the heuristics established by Quinn [1996] to CSI training (cf. also App. B). The experts have to judge whether the goals and tasks of the scenario have been clearly defined ("clear goals and objectives"), whether the training system makes clear the relation between game story and real training ("context meaningful to domain and learner"), or whether the digital game-based training system will support the learner and enhance the learning process ("elicit learner understanding"), for example.

Novel concepts & future ideas: These questions are used to evaluate the novel concepts introduced by the general framework and extended game universe. The category includes five questions, which have not been based on existing heuristics. These questions specifically concern expert feedback on the extended game universe as well as on the general framework for digital game-based training systems. The questions primarily target the evaluation of the experts' ratings on the suitability of both concepts to education. For example, the experts are asked to rate in-game authoring as well as explicit links to real training, which are provided by *OpenCrimeScene* in the form of the digital SLR camera. Questions on future ideas target the rate of the visual review log as a classroom supplement as well as the use of different technologies other than interactive 3D environments.

Following Albion [1999], a rating scale is used as an answering measure for each question. The scale is illustrated in Table 5.1 and was chosen to reconcile the different evaluation goals that go from general assumptions on user knowledge to usability principles.

$--$	$-$	o	$+$	$++$	n/a
very low	low	normal	high	very high	not applicable

Table 5.1.: Rating scale used to evaluate the different questions.

Since it is of particular interest to receive feedback from the actual expert end users of the prototype, the focal point is put on educational usability and usefulness of the system. Since this might result in a number of questions that can hardly be answered by developers, questions can be rated as *not applicable*.

Having mapped the evaluation goals to individual question categories and given the rating measure, the evaluation will be conducted as follows: each expert group has to be handed out the questionnaires as well as an accompanying CD-ROM that contains a demo video and prototype executable.[11] Each group then has to be split in two: one group has to watch the video and answer the questions; the other group has to execute the prototype and answer the questions. Each questionnaire furthermore contains an introductory part that carefully introduces the experts to the goals of the questionnaire and describes how to conduct the evaluation. The results of the evaluation will be discussed in great detail in the next section.

5.4.3 Results of the Evaluation

As was already mentioned, the evaluation was intended to obtain a first feedback from the experts in order to get a certain trend of answers. In order to assess the ratings of the different expert groups, an average value was calculated for each expert group's ratings. It was decided to calculate the arithmetic mean of each rating, even though it is not necessarily an objective representation of averaged values. This is true, especially in the case of statistical anomalies. In such cases, weighted mean would have resulted in more accurate average values. However, to calculate weighted mean, a thorough study would have been required to establish an "objective" weighting factor. Yet since the evaluation

11 The demo video and prototype executable can be found on the accompanying CD-ROM of this dissertation.

was intended to obtain a first feedback of the experts and since statistical anomalies have rarely been found in the answers, accuracy was neglected in favor of prompt evaluation results. Furthermore, occurrences of statistical anomalies will be explicitly pointed out and can also be seen in the complete presentation of results in Appendix D.

To calculate the arithmetic mean, the rating scale was mapped to scalar values ranging from 1 to 5, as illustrated in Table 5.2. If one or more participants chose n/a as their answer only, the arithmetic mean of the remaining answers was calculated. Hence, the number of n/a answers does not influence the result of the calculation, yet it will be explicitly pointed out and discussed for each answer. Moreover, some questions had to be answered by yes or no. These were not averaged, but will be plainly discussed.

$--$	$-$	o	$+$	$++$
very low	low	normal	high	very high
1	2	3	4	5

Table 5.2.: Rating scale used to calculate the arithmetic means of the answers.

In the following, the averaged results of each category will be illustrated and discussed. To better relate the results to the individual questions, each question is added to the illustrations. Questions are abbreviated by Q; questions 1 to n will be referred to as $Q1$ to Qn in the illustrations as well as in the discussion.

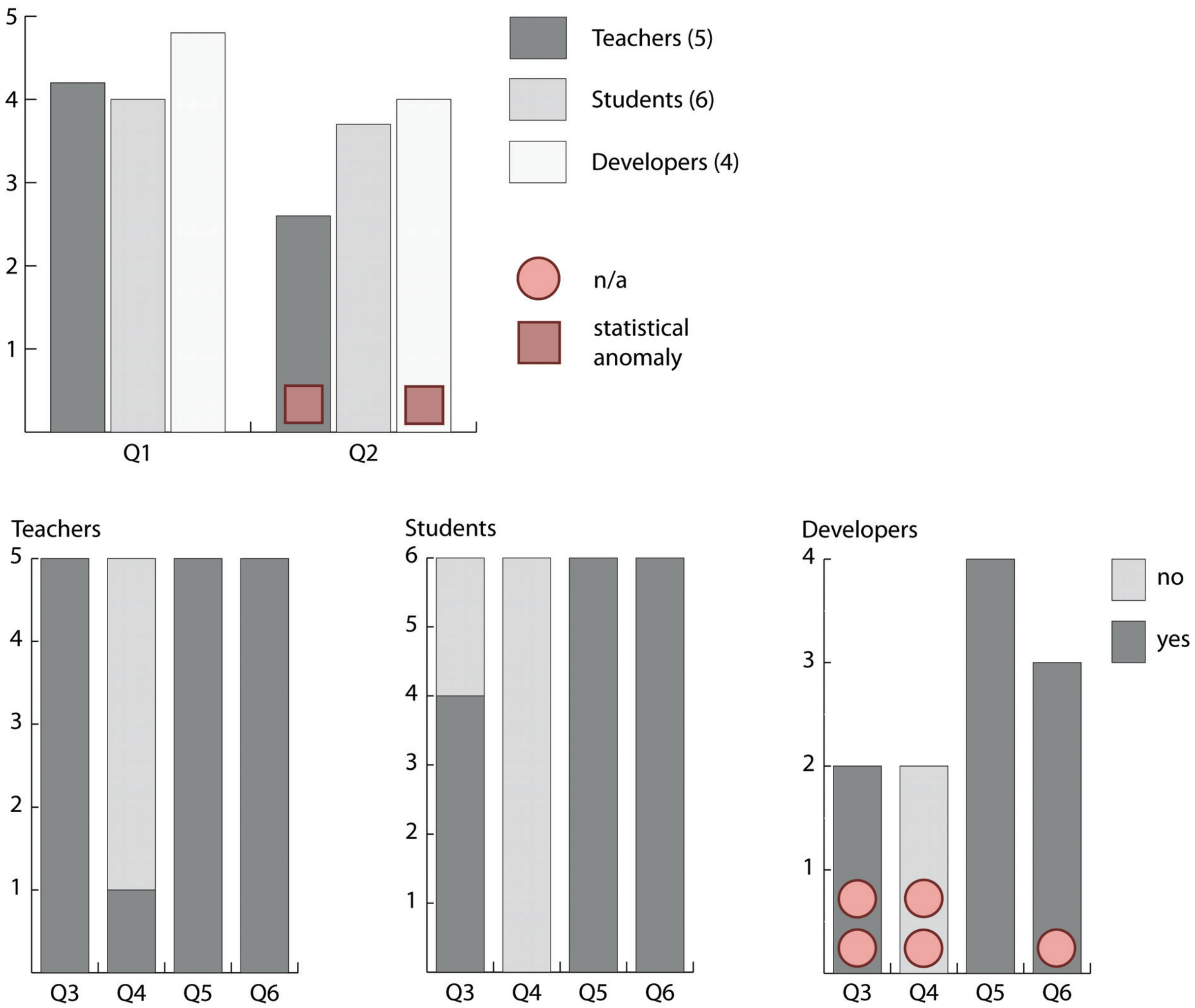
General Questions

In this category, first of all, experts had to give general information concerning age, gender, and membership of the expert group. Accordingly, the expert groups consisted of

- 5 teachers; average age: 46.0 years (ranging from 37 to 58 years); all male
- 6 students; average age: 23.0 years (ranging from 21 to 30 years); 3 female, 3 male
- 4 developers; average age: 28.5 years (ranging from 28 to 30 years); all male.

The results of the questions that had to be rated by the experts are illustrated in Figure 5.31.

Teacher and student experts rated themselves as “highly” experienced with PC software applications, whereas developers naturally go beyond that ($Q1$). In contrast, experience



Q1: How do you rate your PC experience?

Q2: How do you rate your experience with computer games?

Q3: Have you already worked with PC software applications in class?
If so, which software applications did you use?

Q4: Have you already worked with computer games in class?
If so, which games did you use?

Q5: Do you consider digital game-based training systems conducive to training?

Q6: Would you use digital game-based training systems in class on a trial basis?

Figure 5.31.: The graph illustrates the results in the category of general questions.

of computer games is definitely rated higher by student experts than by teacher experts ($Q2$). However, the difference is not remarkable. This might be caused by the fact that female student experts did not rate their familiarity with computer games as high as their male colleagues. Moreover, one teacher expert judged himself to be "very highly" experienced with computer games whereas the other four experts rated themselves as of limited experience ("low") (cf. App. D). Also, one out of four developers rated himself as of very limited experience with computer games ("very low"), whereas the other three experts rated themselves as "very highly" experienced, which resulted in an average value of "highly" experienced developer experts.

The last four questions had to be answered either by yes or no. Questions $Q3$ and $Q4$ asked whether the participant ever had used PC software ($Q3$) or computer games ($Q4$) for lessons. Generally, PC software like *Microsoft Office* applications have been used in class. However, two out of six student experts did not use PC software in class. In contrast, computer games have not been used in class by any expert — except for one. Interestingly, one out of five teacher experts had used the game environments *Game Maker* and *Alice* in class. Unfortunately, no further explanation was provided. The answers given by the developers follow this trend: PC software has been used for teaching purposes at the university, whereas computer games have not. The red circles indicate that two developer experts could not answer this question appropriately because they had no teaching experience.

Finally, the last two results clearly say that all experts consider digital game-based training systems an enhancement to learning ($Q5$) and would use a digital game-based training system in class on a trial basis ($Q6$). The results conform to what has been expected. Use of digital game-based training systems to enhance training is new to almost all teacher and student experts, who are open to the use of such applications in class. Moreover, student experts can be considered to be generally more familiar with computer games than teacher experts. On the other hand, all experts are familiar with common PC software applications and, hence, can be considered as being well acquainted with PC use.

Usability Principles

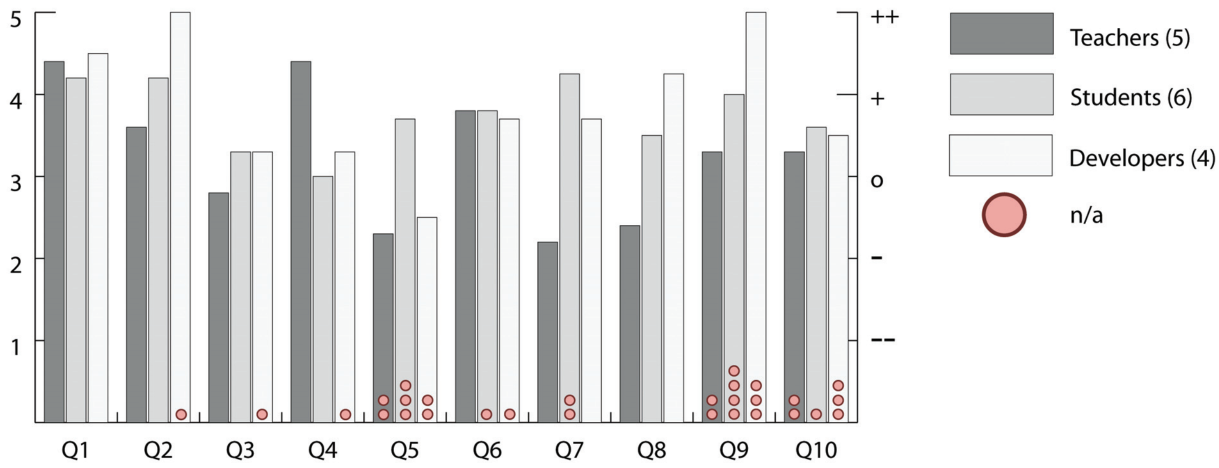
The rating of the usability principles is illustrated in Figure 5.32. At first glance, the illustration shows that various questions could not be clearly answered, which is indicated by several red circles. This is caused by the fact that features like user support in case of system errors ($Q5$), error messages ($Q9$), or a user manual ($Q10$) have not been fully integrated into *OpenCrimeScene* so far. Especially experts who have watched the demo video could not clearly specify how the system responds to user input, for example.

Apart from that, most corresponding values resulted from $Q1$, $Q3$, and $Q6$. These questions concerned the overall system status ($Q1$), freedom of user navigation ($Q3$), and clearness of user interaction ($Q6$). Overall system status and clearness of user interaction have both been rated "high" by most experts, whereas freedom of user navigation has been rated as "normal." This might be caused by the fact that not all of the experts interacted with the prototype but watched the demo video.

Results of $Q2$, $Q4$, $Q7$, and $Q8$ indicate differences in the experts' ratings. Although the overall rating of $Q2$ is beyond "normal," menu items were considered to be "very highly" clear by the developers and "normally" clear by the teacher experts. However, when asked about the program structure in relation to known software applications ($Q4$), teacher experts judged the program to conform "highly" to other applications — in contrast to student and developer experts. Clearness of menu items and program structure are closely related and since the results are not consistent, both aspects will have to be re-investigated. Probably, the question's phrasing was misleading, which might have caused the discrepancy in the teacher ratings.

Question $Q7$ asked the experts to rate whether they could efficiently work with the system. Student experts expected themselves to be "highly" efficient with the system and gave a "high" rate (beyond $+$). In contrast, teachers expected themselves to be less efficient and gave a "low" rate (even below $-$). This might indicate that teachers do not yet feel comfortable with these software applications. On the other hand, the prototype state did not allow teachers to actually work with the system and prepare lessons. Therefore, this question will have to be re-investigated with fully-functioning prototype.

Finally, $Q8$ asked about the system's graphical design, which was "very highly" appreciated by the developers and "highly" as well as "normally" appreciated by student and teacher experts. This might be caused by the fact that developer experts know about the effort that has to be put into graphical software design. In contrast, student and teacher experts are probably familiar with commercial products only, which typically have more financial and manpower resources that can be applied to graphical design aspects.



Q1: During the video/program run, did you principally know which part of the program you were in and what it was about?

Q2: Did you find the displayed menu items clear and understandable?

Q3: How do you rate freedom of user control (navigation, interaction, error handling, program end, ...)?

Q4: How familiar does the general program structure seem compared to applications you already know?

Q5: Please rate how the system supports the user in avoiding mistakes (for instance, when wrongly selecting a scene object)?

Q6: Are the interaction means via mouse or the menu easy to recall and, hence, to reach?

Q7: Please estimate whether you could work with the program efficiently, for example, in order to commit a crime as a virtual criminal?

Q8: Does the graphical program design appeal to you?

Q9: Are the error messages of the program understandable?

Q10: How do you rate the program's manual?

Figure 5.32.: The graph illustrates the results in the category of usability principles.

On the whole, the developer experts rated almost all heuristics to be “normal” and beyond, which indicates that the prototype is adequately crafted so far. This is also true for the ratings given by student experts. Only the teacher gave “low” ratings to several questions. As a consequence, the prototype’s interface ought to be improved, especially in regard to $Q5$, $Q7$, and $Q8$, that is, error handling, working efficiency, and graphical design.

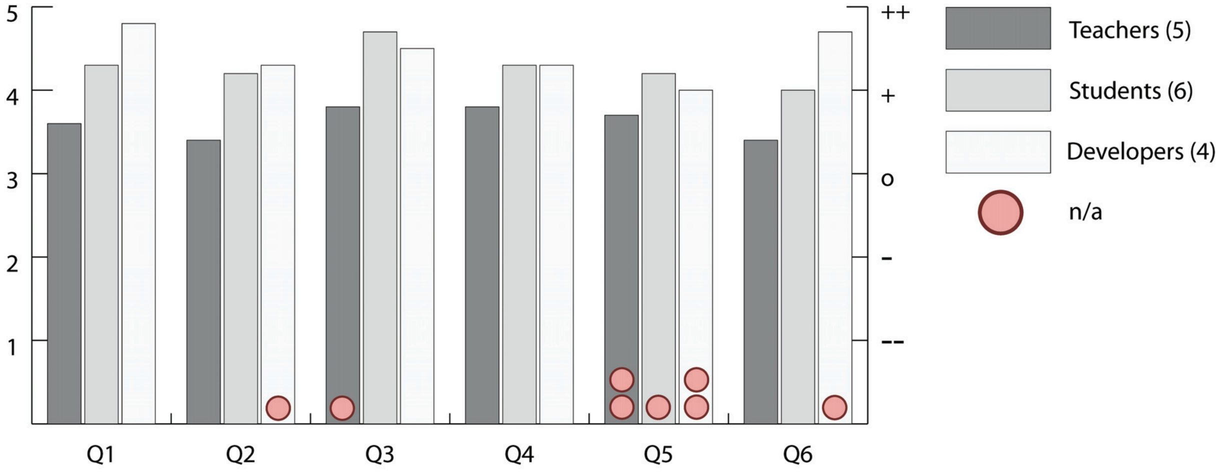
Usefulness for Education

The results of the prototype’s usefulness for education go beyond the rating “normal,” as illustrated in Figure 5.33. This is a very good result in regard to digital game-based training in general and the development of *OpenCrimeScene* in particular. Especially student experts assume the digital game-based training system to be “highly” useful for education and give even higher rates than their teacher counterparts. This conforms to the general assumption that students are usually motivated to use digital training enhancements. It also indicates, however, that teachers do not disapprove of using such applications, as was assumed by several publications on this topic (cf. Sec. 2.3).

Regarding the different questions, all experts understood the goals of the presented scenarios ($Q1$) and could relate the scenario to real training situations ($Q2$). A slight discrepancy occurs between teacher and student experts on this point though. It might be caused by the fact that students are more familiar with computer games and the according data presentations than teachers.

In addition, all experts approved of the cops-&-robbers game story ($Q3$), especially student experts approved “very highly” of the story concept. Furthermore, all experts agreed that digital game-based training systems could enhance CSI training “highly” ($Q4$). Interestingly, most n/a answers were given when asked how to integrate the digital game-based training system into the syllabus ($Q5$). This indicates a strong problem, as was already pointed out by several researchers (cf. Sec. 2.3). When asked about whether the experts would use *OpenCrimeScene* to support CSI training ($Q6$), all experts agreed on it, even though the ratings range between “normal” and “very high.”

Regarding the contribution of the developers, their ratings have been above the other two expert group ratings in most cases. This might be caused by the developers’ general familiarity with software applications. As a consequence, developers are likely to trust in



Q1: Did you understand the task and goal of the respective scenario?

Q2: Could you establish a relationship between the program context and real training?

Q3: Do you like the idea of doing training by playing "cop vs. robber"?

Q4: Can the training game enhance studying and make it more effective, for example, in regard to reinforcing knowledge?

Q5: Can the system be intergrated into the syllabus? Please give a comment how this could be done.

Q6: Would you like to use such a training game for your studies?

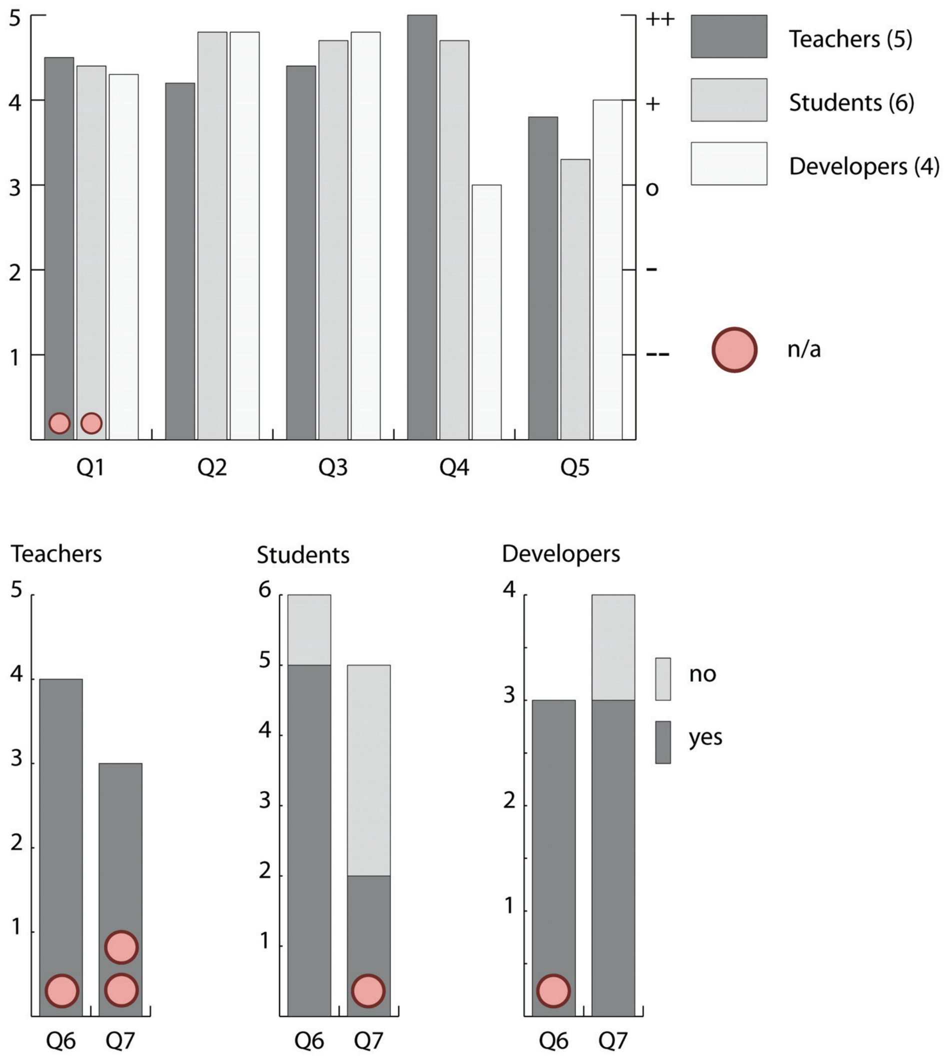
Figure 5.33.: The graph illustrates the results in the category of usefulness for education.

the usefulness of such applications. Teacher experts gave lowest ratings, which could be put down to their lack of experience with these software applications.

Novel Concepts & Future Ideas

The results of the questions on novel concepts & future ideas are illustrated in Figure 5.34. Questions $Q1$ to $Q5$ concern novel concepts, whereas $Q6$ and $Q7$ concern future ideas that have to be answered with either yes or no.

The ratings given by the different expert groups correlate in most cases except for $Q4$ and $Q7$. Question $Q4$ asked the experts to rate if the presentation of training tools should exactly conform to reality, which is rated "normal" by most developers and "very much" by teacher and student experts. Obviously, realistic data presentation is required in a digital game-based training situation, however, further investigation would be required



Q1: How do you rate the fact that the teacher acts as a game player?

Q2: How do you rate the fact that different scenarios can be created as part of the game?

Q3: How do you rate the fact that the system provides real training tools such as the virtual SLR camera to assess the virtual crime scene?

Q4: Would it be an advantage if the presentation of training tools exactly matched reality?

Q5: Would you consider it a benefit if the student acted as the criminal?

Q6: Would you use a visual review log for classroom discussions? Please give a comment.

Q7: Would you consider smart environments conducive to training?

Figure 5.34.: The graph illustrates the results in the category of novel concepts & future ideas.

in regard to non-photorealistic depiction of training tools, as was already mentioned in Section 4.2. Bearing this is mind, developers might have given a lower rating to this question. In a second study, two different visualization styles should be used and empirically assessed considering understanding and learning support.

Question $Q7$ asked the experts to rate whether different technologies other than interactive 3D environments, in particular smart environments, would further support training.[12] This was "very highly" approved of by teachers and lesser approved of by students ("low" to "normal"). Students were either "very highly" or "very low" approving of the idea of smart environments (cf. App. D). Some students commented on the question by saying that the training situation was too unrealistic. Other students said that it would only be partially useful in training.

Apart from these two ratings, the other ratings correlated. All experts liked the novel concept of addressing the teacher as an active player and gave "high" rates ($Q1$). The teacher experts approved of this idea most, which backs up the concept of the extended game universe developed in this dissertation and underlines its necessity.

The authoring aspect that was introduced as the main feature of the teacher player also received "high" ratings by all experts, although the teacher expert ratings were lowest. This was caused by the fact that one teacher abstained from rating $Q1$ and gave a "normal" rate to $Q2$, whereas the other teacher ratings remain equal for $Q1$ and $Q2$ (cf. App. D).

Besides, "high" ratings were also given to the fact that authentic training tools are provided by *OpenCrimeScene* like, for example, the virtual SLR camera ($Q3$). Question $Q5$ concerned the idea of letting the student player act as robber. The intention behind this question was to support different learning strategies. If the student could act as robber, a constructivist approach could be pursued by the training game, for instance. It was especially approved of by developer experts and more approved of by teacher experts than by student experts. Yet the overall rating ranges remained between "normal" and "high." Probably, the question should have been phrased more explicitly and should have pointed out the intention of different training strategies.

Finally, $Q6$ concerns the generation of a visual review log and its integration into classroom discussion. This was "very highly" approved of by all experts. However, one

12 A smart environment was outlined to the interviewees as a real room that contains additional computer technology like laptops, multiple screens, sensors, etc., which automatically supports the user and could be integrated into real training.

student rated the visual review log integration “very low,” arguing that individual training assessment should not be discussed in class. Yet the comment suggests that the student’s focus was not put on the visual review log itself, but rather on the difficulty of discussing his or her training progress in class. In regard to the other ratings, it can be assumed that the visual review log is “very highly” required to support digital game-based training systems as educational supplements because it directly links the digital game-based training to real education.

Based on the discussion of results of the system evaluation, a conclusion will be drawn subsequently.

5.4.4 Conclusion

The system evaluation of *OpenCrimeScene* has led to positive results for the prototype. In particular, the contributions of this dissertation to digital game-based training and the development of digital game-based training systems have been highly valued by all experts, indicating their relevance for digital game-based training. General assumptions that had been made in this dissertation and upon which most novel concepts were built could be proved in the evaluation: teachers are not as familiar with computer games as students, and use of digital game-based training systems is not common in CSI education, but is welcomed by both groups.

The usability principles have generally been met by the prototype, although several improvements have to be made like proper error handling. Especially teacher experts gave several low ratings to the system interface. However, the system is just a first prototype and, thus, lower ratings of interface and provided interaction means seem to be natural. Most student experts and all developer experts generally gave higher ratings. This might indicate that teachers are still not as familiar with interactive 3D environments like digital game-based training systems as the other two groups. Question $Q2$ on familiarity with computer games from the category of general questions might have been misleading by giving the impression that the teachers’ familiarity with computer games was not remarkably less than the students’ familiarity with computer games (see Fig. 5.31). The range of computer game types is broad. It is likely that the question would have led to a different result if it had indicated familiarity with interactive 3D computer games. Similar discrepancies in the results have already been mentioned in the discussion above and show that at least some questions lacked an exact phrasing. These have to be re-investigated in a future study to better understand the different ratings on usability principles.

Apart from that, particularly *OpenCrimeScene's* usefulness for education as well as the novel concepts represented by the prototype have been highly valued by all experts. Even though most teacher experts were a bit more restrained compared to their student counterparts when rating usefulness for education, they strongly embraced novel concepts like the teacher player, in-game authoring, and explicit links to education provided by authentic data presentation and a visual review log. Developer ratings were of particular value in the context of usability principles, but also provided an independent view on the other three categories. In general, all experts welcomed the idea of using *OpenCrimeScene* in class to enhance CSI education.

In regard to future steps, the present system evaluation has to be used as a starting point to improve *OpenCrimeScene* and should serve as a basis for an extensive usability testing to empirically assess the prototype. This said, the end of the workflow schedule has been reached. The next section summarizes the present chapter and points towards the final summary of this dissertation.

5.5 Summary

This chapter presented and discussed the development and implementation of the digital game-based training system *OpenCrimeScene* by passing through the different stages of the workflow schedule. Consequently, the chapter went into establishing the user working profile for CSI training first and led on to developing digital game-based training content. The actual implementation of *OpenCrimeScene* was presented next, which also became the largest part of the chapter due to its complexity. Based on the implementation, a system evaluation was conducted, which proved the prototype's usefulness for CSI training as well as the relevance of the concepts introduced in this dissertation to digital game-based training.

The implementation of *OpenCrimeScene* shows that the general framework for digital game-based training systems introduced in this dissertation serves well as a basis for an actual training system. The prototype comprises all aspects of the general framework like in-game authoring, training, and reviewing training sessions as well as generating a visual review log either as implementation or as conceptualization. Since these were put in the context of CSI training, several research questions arose as a result of the specific training domain. Such questions encompassed authentic fingerprint visualization as well as realistic trace transfer or valid evidence collection. Hence, common approaches to data (re)presentation and human-computer-interaction had to be re-considered in the light of CSI training.

The broad scope of this chapter also shows that the development of a digital game-based training system is fairly extensive, even though the scope of the prototype had been limited. As was already noted, eighteen students contributed to *OpenCrimeScene* as part of student assignments. The large number of contributions underlines the fact that considerable demands are put on design and implementation of a digital game-based training system. It also underlines the fact that the development of a digital game-based training system supplies numerous research topics which should be investigated.

Chapter 6

Concluding Remarks

The analysis of the state of the art and, in particular, of the historical development of educational software on the one hand and computer games on the other hand, has shown that both research threads have long been pursued almost separately, even though they continuously affected each other at least on a technical level. The effects mainly consisted of integrating game elements into training environments for enhancement purposes and transferring the success of computer game playing to learning situations. In this sense, the development of serious games, which emerged only recently and were considered as the "holy grail" for digital game-based learning, comes naturally — yet serious games do not unite educational demands with computer game playing strategies to equal parts. Consequently, digital game-based learning strategies still have not made full entrance into education.

This dissertation has made a contribution to better integrating digital game-based learning strategies into education by approaching the problem from a technical side. A general framework for digital game-based training systems was introduced that addresses educational situations and provides a solution for supporting all participants of education. This final chapter summarizes the results and contributions made during the scope of this dissertation and outlines future work that springs from the presented developments.

6.1 Summary

This dissertation presents three main results: first, a general framework for digital game-based training systems has been developed. The framework is accompanied with a practical workflow schedule that is tailor-made for implementing digital game-based training systems. Second, the applicability of the general framework to digital game-based training systems has been verified by the implementation of a novel digital game-based training system for CSI training called *OpenCrimeScene*. The implementation exemplifies how to use the workflow schedule and can thus serve as a demonstration for other digital game-based training systems. Third, specific solutions have been developed in the context of the general framework and *OpenCrimeScene*, which can also be used individually for other purposes.

General Framework & Workflow Schedule

The general framework is based on the *extended game universe*, which schematically represents educational situations in a game context. Educational situations and game playing situations were closely examined and compared to each other in the forms of the *educational universe* and *game universe* to set up the extended game universe. Unlike these schematic representations, the general framework for digital game-based training systems represents educational situations on a technical basis. It consists of an *authoring component*, a *training component*, and a *reviewing component*, which address teaching situations, learning situations, and communication situations between teacher and student, respectively.

In addition, the general framework introduces a *novel game story concept* to better acquaint teachers and students with the technology of digital game-based training systems. The story concept proposes to split the traditional game story into two parts: the *back-story* and the *main-story*. Both have to be implemented by the authoring and training components, respectively. The back-story addresses teachers as game players. To acquaint them with digital game-based training systems, the interaction paradigm of *in-game authoring* was developed. In-game authoring turns content authoring into a game playing experience: while playing the back-story, teachers can specify the main-story. The main-story addresses students as game players and introduces the interaction paradigm of *in-game training*. In-game training aims at supplementing learning situations and reassuring students that the training content is relevant for education. Therefore, authentic data presentation as well as the integration of existing learning material into the training system is required. The *close dependency* between both stories contributes to communication and interaction situations between teacher and student in real education.

The introduction of back-story and main-story tackles several well-known problems of digital game-based training. First of all, unfamiliarity with computer games, which is generally attributed to teachers, has often hindered the integration of serious games into education. The back-story concept presents a first way to overcome this problem because it explicitly invites teachers to use digital game-based training systems for their lessons and to enjoy digital game-based training. Moreover, the possibility to author training content supports teaching situations and, hence, the educational role of teachers.

Apart from the teaching aspect, the general framework also gives *new input to the training aspect*. The introduction of two game player roles, which can either be opponents or accomplices, will most probably enhance students' motivation to learn. Although it may sound placative, "beat the teacher" might provide some extra motivation to engage with the learning content. The fact that the learning content has been specified by the teacher may also prevent students from discrediting the educational use of digital game-based training systems. Hence, the authoring and training component provide a way to *integrate digital game-based training systems into education by motivating teachers and students* to use digital game-based training systems. In contrast, the reviewing component pursues a different strategy of integration. The novel concept of the *visual review log* can be used for classroom discussions and specifically aims at *adapting the new technology to traditional educational situations*. This goal of balancing novelty and tradition is considered crucial property of the general framework because it provides a method of smoothly transferring the ideas of digital game-based learning to education.

The general framework and the novel concepts introduced put several demands on the implementation of a digital game-based training system. Especially, authoring of training content as well as reviewing training sessions pose challenges to research and require further investigation. However, they are essential to pursue the idea of digital game-based learning and training. As a result of discussing such implementation issues, a *workflow schedule* was established. The workflow schedule is related to instructional system design, yet it is specifically formed to the requirements of digital game-based training systems. The schedule was successfully applied to the implementation of a digital game-based training system for CSI training: *OpenCrimeScene*.

OpenCrimeScene The implementation of the *OpenCrimeScene* prototype verified *the applicability of the general framework* to an actual implementation as well as *the validity of the workflow schedule* to developing digital game-based training systems. *OpenCrimeScene* has been presented and discussed in great detail in the previous chapter. Therefore, only a few comments will be given here. First of all, the prototype *reproduces common educational situations of CSI training* in which police teachers prepare different crime scene situations and students have to train investigation procedures. This was mapped to a *cop-&-robber game story* in which the teacher can act as a robber and the student can act as a cop. The back-story provides means for committing a crime, whereas the main-story provides means for investigating the crime, accordingly. So far, reviewing functionality is pending, but a *first solution to the visual review log* already exists together with further visualization concepts. On the whole, the prototype fruitfully demonstrates how digital crime scene investigation training could be done taking the general framework as an implementation basis. A *system evaluation* finally confirmed the usefulness of *OpenCrimeScene* for CSI training and substantiated that the general framework is relevant when applying digital game-based learning to real world education. Several concepts developed and implemented as part of the general framework or *OpenCrimeScene* can also be applied to other domains.

Specific Solutions The game story concept of back-story and main-story as well as the interaction paradigm of in-game authoring could well be applied to computer game playing. In particular, game mastering techniques required for role playing games will gain from in-game authoring. Apart from that, a combination of modding and in-game authoring could be used to enhance story-based computer game. In addition to creating level content, in-game authoring could be used to specifying story content. Furthermore, the concept of the reviewing component and visual review log could be extended to computer games.

The development of *OpenCrimeScene* resulted in many specific solutions. To name a few, *authentic evidence visualization*, *implicit trace transfer*, and *virtual crime scene photography* were introduced here. The underlying concepts could be used to add realistic features to computer games. For instance, the virtual SLR camera could be used as a stand-alone training system or be adapted to realistic simulation of lens effects in 3D virtual environments.

6.2 Future Work

All through this dissertation the necessity of a general framework for digital game-based training systems has been maintained. It is assumed that it will pave the way for the successful integration of digital game-based training systems and digital game-based learning into education. Further developments will be necessary to continue this process.

In the general context of education, it should be examined if different learning strategies could be pursued with the game story concept of back-story and main-story. As was already mentioned, in-game authoring could prove useful when following a constructivist/constructionist learning strategy. In addition, it would be very interesting to investigate if the story's visualization and rendering style influences learning and training processes. Collaboration with researchers from interactive drama as well as from pedagogy and psychology will be helpful to investigate these issues.

The visual review log will raise additional research questions. Throughout this dissertation it was asserted to be supportive to teacher-student-communication in real educational situations. One way to visually summarize user interaction was provided by *OpenCrimeScene*, yet the question remains if there is a general approach to generating visual review logs from interactive 3D environments. A general framework ought to be developed that turns the different steps of data specification, data assessment, and data visualization into a technical representation. Primarily, a general task model will be required that allows for defining and specifying data. It has to take into account internal as well as external data. Moreover, automatic and parameterizable data assessment strategies have to be developed, which allow users to specify data of interest.

Data specification and assessment alone will comprise many research questions. Data visualization will raise further questions. These ought to concern how to depict user interaction and even user intention, which is of particular interest in education in regard to task accomplishment. In the context of task accomplishment, it is likely that certain user interaction patterns can be examined that predict user intention. Apart from that, it will be worth investigating if different training aspects require different visualization styles. For example, if the overall training procedure is of interest for classroom discussion, the visual review log might have to emphasize course-based aspects of the training like different character positions. On the contrary, if task accomplishment is of interest for classroom discussion, the visual review log might have to emphasize event-based aspects of the training that include sequences of user-object-interaction. Hence, different input data have to be rendering and stylized, which puts demands on the implementation.

In regard to *OpenCrimeScene*, several basic features will have to be added to the prototype like collision detection or physics, as was already discussed. Yet the prototype provides further research topics. First of all, evidence collection and processing raise interesting research topics that concern deployment of texture mapping and shader programming. Both techniques provide various possibilities to efficiently visualize data and could be used to visualize evidence processing. Therefore, functionality has to be assigned to the forensic tools. In this context, the idea of the *virtual-hand-editor* provides an interesting topic to realistically simulate trace transfer to scene objects. Finally, also the reviewing component ought to be extended to comprise of general functionality, in respect to the considerations made above.

The system evaluation of *OpenCrimeScene* substantiated the usefulness of the general framework for integrating digital game-based training systems into education when applied to interactive 3D environments. Other technologies, in particular, the novel concept of *smart meeting rooms*, should be taken into future consideration. The Graduiertenkolleg MuSAMA, which is held at the University of Rostock, is investigating aspects of that field of research.[1] A smart meeting room is a real environment that contains a multitude of interconnected sensor and computer devices. These range from desktop PCs to digital projectors, motion trackers, laptops, cell phones, multiple display screens, and many more. The devices automatically track the user's behavior in the smart environment, analyze it, and predict what kind of services the user might require in the current situation. Yet smart meeting rooms often lack practical application scenarios.

The general framework that was developed in this dissertation could be used as a generic control basis for smart meeting rooms. Authoring, training, and reviewing components could be dynamically distributed to different computer devices and output screens, in respect to the current training situation. A training scenario is well-suited to predict user intention because it is generally determined by a set of tasks users have to accomplish. The visual review log concept could be extended to extract user intention patterns. These could, in turn, be used to fine-tune analysis and prediction methods of the smart meeting room. In the context of *OpenCrimeScene*, teachers could prepare the smart meeting room to present additional information during crime scene investigation. When students make mistakes during the investigation, help texts or quizzes could be automatically triggered and displayed on screen devices close to the students' field of view. A smart meeting room could also be used to unite different lessons like practical CSI training and legal issues. Thus, smart meeting rooms are uniquely suited to combine digital training and practical training into *smart training rooms*.

The aspects discussed in this section present a starting point for future investigation. Further questions and research topics will hopefully be raised based upon them and support communication between researchers and end users.

1 For further information see [http://www.informatik.uni-rostock.de/musama.html] (March 22, 2009).

Bibliography

Ahl, D. H. (1984). The First Decade of Personal Computing. *Creative Computing*, 10(11):30. [http://www.atarimagazines.com/creative/v10n11/30_The_first_decade_of_perso.php] (March 22, 2009).

Albion, P. R. (1999). Heuristic Evaluation of Educational Multimedia: From Theory to Practice. The 16th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (Ascilite '99). [http://www.ascilite.org.au/conferences/brisbane99/papers/albion.pdf] (March 22, 2009).

Ally, M. (2004). *Theory and Practice of Online Learning*, chapter "Foundations of Educational Theory for Online Learning". Athabasca University, Athabasca, AB, Canada.

Anderson, C. A. (2000). Violent Video Games Increase Violence and Aggression. U.S. Senate Committee on Commerce, Science, and Transportation hearing on "The Impact of Interactive Violence on Children". [http://www.psychology.iastate.edu/faculty/caa/abstracts/2000-2004/00Senate.html] (March 22, 2009).

Anderson, C. A. (2003). Violent Video Games: Myths, Facts, and unanswered Questions. *Psychological Science Agenda: Science Briefs*, 16(5):1–3. [http://www.apa.org/science/psa/sb-anderson.html] (March 22, 2009).

Ardito, C., Costabile, F., Marsico, M. D., Lanzilotti, R., Levialdi, S., Roselli, T., and Rossano, V. (2006). An Approach to Usability Evaluation of E-learning Applications. *Universal Access in the Information Society*, 4(3):270–283.

Ark, W., Dryer, D. C., Selker, T., and Zhai, S. (1998). Representation Matters: The Effect of 3D Objects and a Spatial Metaphor in a Graphical User Interface. In *People and Computers, Proceedings of HCI'98*, pages 209–219. Springer.

Ashcroft, J., Daniels, D. J., and Hart, S. V. (2004). Crime Scene Investigation: A Reference for Law Enforcement Training. National Institute of Justice, Washington, DC.

Bartle, R. A. (2003). *Desiging Virtual Worlds*. New Riders, Indianapolis, IN, USA.

Baumgartner, P. and Payr, S. (1999). *Lernen mit Software*. Studien-Verlag, Innsbruck, Germany, 2nd edition.

Baumgärtner, S., Ebert, A., Deller, M., and Agne, S. (2007). 2D meets 3D: A Human-Centered Interface for Visual Data Exploration. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, pages 2273–2278, New York, NY, USA. ACM.

Behr, K.-M. (2008). *Die Computerspieler — Studien zur Nutzung von Computergames*, chapter "Kreative Spiel(weiter)entwicklung: Modding als Sonderform des Umgangs mit Computerspielen", pages 193–207. VS Verlag für Sozialwissenschaften.

Benford, S., Bederson, B. B., Åkesson, K.-P., Bayon, V., Druin, A., Hansson, P., Hourcade, J. P., Ingram, R., Neale, H., O'Malley, C., Simsarian, K. T., Stanton, D., Sundblad, Y., and Taxén, G. (2000). Designing Storytelling Technologies to Encourage Collaboration Between Young Children. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'00)*, pages 556–563, New York, NY, USA. ACM.

BinSubaih, A., Maddock, S., and Romano, D. (2008). A Practical Example of the Development of a Serious Game for Police Training. *Handbook of Research on Effective Electronic Gaming in Education*. In press.

Bitzer, D. L., Blomme, R. W., Sherwood, B. A., and Tenczar, P. (1970). The PLATO System and Science Education. Technical Report X–17, Computer-based Education Research Laboratory, University of Illinois, Urbana, IL, USA.

Bitzer, D. L., Braunfeld, P. G., and Lichtenberger, W. (1961). PLATO: An Automatic Teaching Device. *IRE Transactions on Education*, E-4:157–161.

Bitzer, D. L., Lichtenberger, W., and Braunfeld, P. G. (1962). PLATO II: A Multiple-Student Computer Controlled Teaching Machine. *Programmed Learning and Computer-based Instruction*, pages 205–216.

Bitzer, D. L., Lyman, E. R., and Jr., J. A. E. (1966). The Uses of Plato: A Computer Controlled Teaching System. In *Proceedings of the 1966 Clinic on Library Applications of Data Processing*, number 4 in Clinic on Library Applications of Data Processing, pages 34–46, Illinois, , USA. University of Illinois at Urbana-Champaign.

BIU (2009). Marktzahlen Computer- und Videospiele — Gesamtjahr 2008. Bundesverband Interaktive Unterhaltungssoftware e.V., Fact sheet, Berlin. [http://www.biu-online.de/fileadmin/user/dateien/BIU_Marktzahlen_Gesamtjahr_2008.pdf] (March 22, 2009).

Bluemel, E., Hintze, A., Schulz, T., Schumann, M., and Stuering, S. (2003). Virtual Environments for the Training of Maintenance and Service Tasks. In *Proceedings of the 35th Conference on Winter Simulation (WSC 2003)*, pages 2001–2007. Winter Simulation Conference.

Blunt, R. D. (2006). *A Casual-Comparative Exploration of the Relationship between Game-Based Learning and Academic Achievement: Teaching Management with Video Games*. Dissertation, Walden University. [http://www.waldenu.edu/] (March 22, 2009).

Bogost, I. (2006). *Unit Operations — An Approach to Videogame Criticism*. The MIT Press, Cambridge, MA, USA.

Bouras, C., Igglesis, V., Kapoulas, V., Misedakis, I., Dziabenko, O., Koubek, A., Pivec, M., and Sfiri, A. (2004). Game-Based Learning Using Web-Technologies. *Journal of Intelligent Games and Simulations*, 3(2):67–84.

Bower, G. H. and Hilgard, E. R. (1983). *Theorie des Lernens*. Klett Verlag, Stuttgart, Germany, 5th edition.

Brand, S. (1972). Spacewar. *Rolling Stone*. [http://www.wheels.org/spacewar/stone/rolling_stone.html] (March 22, 2009).

Brennecke, A., Panzer, C., and Schlechtweg, S. (2008a). vSLRcam — Taking Pictures in Virtual Environments. *The Journal of WSCG*, 16(1–3):9–16.

Brennecke, A., Schlechtweg, S., and Schumann, H. (2008b). Game-Based Training in an Interactive Environment by means of OpenCrimeScene. In *Theory and Practice of Computer Graphics 2008 — Eurographics UK Chapter Proceedings*, pages 167–170. University of Manchester, UK, Eurographics Association.

Brennecke, A., Schlechtweg, S., and Strothotte, T. (2007). OpenCrimeScene Review Log: Interaction Log in a Virtual Crime Scene Investigation Learning Environment. In *Proceedings of the 2nd International Conference on Computer Graphics Theory and Applications (GRAPP 2007)*, pages 185–190. INSTICC Press.

Brennecke, A. and Schumann, H. (2009). A General Framework for Digital Game-Based Training Systems. In *IADIS Game and Entertainment 2009*. Accepted as full paper.

Brosch, T. (2006). Blutspurenmodellierung. Term paper. Otto-von-Guericke University, Department of Simulation and Graphics, Magdeburg, Germany.

Bruckman, A. (1997). *MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids*. Dissertation, School of Architecture and Planning, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA.

Bruns, B. and Gajewski, P. (2002). *Multimedials Lernen im Netz: Leitfaden für Entscheider und Planer*. Springer-Verlag, Berlin, Germany, 3^{rd} edition.

Capps, M. (2002). Collaborative Virtual Entertainment: (If You Can't Beat 'em, Join 'em?). In *Proceedings of the 4^{th} International Conference on Collaborative Virtual Environments (CVE '02)*, pages 63–63, New York, NY, USA. ACM Press.

Capps, M., McDowell, P., and Zyda, M. (2001). A Future for Entertainment-Defense Research Collaboration. *IEEE Computer Graphics and Applications*, 21(1):37–43.

Cassandras, C. G. and Lafortune, S. (1999). *Introduction to Discrete Event Systems (The International Series on Discrete Event Dynamic Systems)*. Springer, Dordrecht, The Netherlands.

Champion, E. (2005). Meaningful Interaction in Virtual Learning Environments. In *Proceedings of the Second Australasian Conference on Interactive Entertainment*, pages 41–44, Sydney, Australia, Australia. Creativity & Cognition Studios Press.

Chen, S. and Michael, D. (2005). Proof of Learning: Assessment in Serious Games. *Gamasutra*. [http://www.gamasutra.com/features/20051019/chen_01.shtml] (March 22, 2009).

Chittaro, L. and Ieronutti, L. (2004). A Visual Tool for Tracing Users' Behavior in Virtual Environments. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2004)*, pages 40–47, New York, NY, USA. ACM Press.

Clages, H. (1997). *FH-Schriftenreihe Polizei - Kriminalistik*. Richard Boorberg Verlag, Stuttgart.

Clarke, D. and Duimering, P. R. (2006). How Computer Gamers Experience the Game Situation: A Behavioral Study. *Computers in Entertainment (CIE)*, 4(3):1–23.

Cockburn, A. (2004). Revisiting 2D vs 3D Implications on Spatial Memory. In *Proceedings of the Fifth Conference on Australasian User Interface (AUIC '04)*, pages 25–31, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Cockburn, A. and McKenzie, B. (2002). Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments. In *Proceedings of the SIGCHI Conference* *on Human Factors in Computing Systems (CHI '02)*, pages 203–210, New York, NY, USA. ACM.

Cohen, S. (1984). *ZAP! The Rise and Fall of Atari*. McGraw-Hill, New York, NY, USA.

Corradini, A., Mehta, M., Bernsen, N.-O., and Charfuelan, M. (2005). Animating an Interactive Conversational Character for an Educational Game System. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*, pages 183–190, New York, NY, USA. ACM Press.

Crawford, C. (1984). *The Art of Computer Game Design: Reflections of a Master Game Designer*. McGraw-Hill/Osborne Media, web edition. [http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html] (March 22, 2009).

Curtis, P. and Nichols, D. A. (1993). MUDs Grow Up: Social Reality in the Real World. Technical report, Xerox PARC, Palo Alto, CA, USA. [ftp://ftp.lambda.moo.mud.org/pub/MOO/papers/MUDsGrowUp.txt] (March 22, 2009).

Davies, N., Mehdi, Q., and Gough, N. (2004). Virtual Crime Scene Reconstruction with Integrated Animated Characters. In *Proceedings of the 18th European Simulation Multiconference (ESM 2004)*, pages 338–393, Magdeburg, Germany. The European Council for Modelling and Simulation (ECMS).

de Oliveira, J. C., Shirmohammadi, S., and Georganas, N. D. (1999). Collaborative Virtual Environment Standards: A Performance Evaluation. In *Proceedings of the 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications (IEEE DIS-RT '99)*, Washington, DC, USA. IEEE Computer Society.

de Villiers, M. R. (2007). An Action Research Approach to Design, Development, and Evaluation of an Interactive E-Learning Tutorial in a Cognitive Domain. *Journal of Information Technology Education*, 6:455–479.

Desurvire, H., Caplan, M., and Toth, J. A. (2004). Using Heuristics to Evaluate the Playability of Games. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, pages 1509–1512, New York, NY, USA. ACM.

Dondlinger, M. J. (2007). Educational Video Game Design: A Review of the Literature. *Journal of Applied Educational Technology*, 4(1):21–31.

Ebert, D. S., Sousa, M. C., Gooch, A., and Stredney, D. (2005). Computer-Generated Medical, Technical, and Scientific Illustration. In *ACM SIGGRAPH 2005 Courses (SIGGRAPH '05)*, page 1, New York, NY, USA. ACM.

Egenfeldt-Nielsen, S. (2005). *Beyond Edutainment — Exploring the Educational Potential of Computer Games*. Dissertation, IT-University of Copenhagen, Copenhagen, Denmark.

Egenfeldt-Nielsen, S. (2006). Overview of Research on the Educational use of Video Games. *Digital Kompetanse — The Nordic Journal of Digital Literacy*, 1:184–213.

El-Nasr, M. S. and Smith, B. K. (2006). Learning Through Game Modding. *Computers in Entertainment*, 4(1):7.

Elliot, J., Adams, L., and Bruckman, A. (2002). No Magic Bullet: 3D Video Games in Education. In *Proceedings of International Conference of the Learning Science (ICLS'02)*, Seattle, WA, USA.

Elliot, J. L. (2005). *AquaMOOSE 3D: A Constructionist Approach to Math Learning Motivated by Artistic Expression*. Dissertation, Georgia Institute of Technology, Atlanta, GA, USA.

Elliott, J. and Bruckman, A. (2002). Design of a 3D Interactive Math Learning Environment. In *DIS '02: Proceedings of the 4th conference on Designing interactive systems*, pages 64–74, New York, NY, USA. ACM.

Entertainment Software Association (2008). Essential Facts about the Computer and Video Game Industry. Fact sheet. [http://www.theesa.com/facts/pdfs/ESA_EF_2008.pdf] (March 22, 2009).

Ertmer, P. A. and Newby, T. J. (1993). Behaviorism, Cognitivism, Constructivism: Comparing Critical Features from an Instructional Design Perspective. *Performance Improvement Quarterly*, 6(4):50–70.

Fanderclai, T. L. (1995). MUDs in Education: New Environments, New Pedagogies. *Computer-Mediated Communication Magazine*, 2(1):8. [http://www.ibiblio.org/cmc/mag/1995/jan/fanderclai.html] (March 22, 2009).

FAS and ESA (2006). Harnessing the Power of Video Games for Learning. Summit on Educational Games, Federation of American Scientists (FAS) and Entertainment Software Association (ESA), NW, Washington, DC, USA. [http://www.fas.org/programs/ltp/policy_and_publications/summit/SummitonEducationalGames.pdf] (March 22, 2009).

Fielding, D., Logan, B., and Benford, S. (2006). Balancing the Needs of Players and Spectators in Agent-based Commentary Systems. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 996–998, New York, NY, USA. ACM Press.

Flowers, A., Magerko, B., and Mishra, P. (2006). Gamemasters and Interactive Story: A Categorization of Storytelling Techniques in Live Roleplaying. *Futureplay*.

Frazer, A., Argles, D., and Wills, G. (2007). Is Less Actually More? The Usefulness of Educational Mini-games. *IEEE International Conference on Advanced Learning Technologies*, 0:533–537.

Frenkel, K. (1994). A Conversation with Alan Kay. *interactions*, 1(2):13–22.

Friedman, D. A., Feldman, Y. A., Shamir, A., and Dagan, T. (2004). Automated Creation of Movie Summaries in Interactive Virtual Environments. In *Proceedings of the IEEE Virtual Reality Conference (VR 2004)*, pages 191– 200, Los Alamitos, CA, USA. IEEE Computer Society.

Fulton, S. (2007). The History of Atari: 1971–1977. *Gamasutra*. [http://www.gamasutra.com/view/feature/2000/the_history_of_atari_19711977.php?page=1] (March 22, 2009).

Fun, I. L. S., Sunar, M. S., and Bade, A. (2004). Non-Photorealistic Outdoor Scene Rendering: Techniques and Application. *Computer Graphics, Imaging and Visualization, International Conference on*, 0:215–220.

Gasch, S. (2005). Alan Kay. *Web article*. [http://ei.cs.vt.edu/~history/GASCH.KAY.HTML#4] (March 22, 2009).

Gee, J. P. (2003). *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan, New York, NY, USA, 1st edition.

Gibson, S. and Howard, T. (2000). Interactive Reconstruction of Virtual Environments from Photographs, with Application to Scene-of-Crime Analysis. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST 2000)*, pages 41–48, New York, NY, USA. ACM Press.

Goldman, D. B., Curless, B., Salesin, D., and Seitz, S. M. (2006). Schematic Storyboarding for Video Visualization and Editing. *ACM Trans. Graph.*, 25(3):862–871.

Gooch, B. and Gooch, A. (2001). *Non-Photorealistic Rendering*. AK Peters Ltd., Nattick, MA, USA.

Good, J., Romero, P., Robertson, J., and du Boulay, B. (2006). Authoring as Acting: Exploring Embodied Interaction in Game Authoring Environments for Children. Project proposal, online publication. [http://www.informatics.sussex.ac.uk/research/projects/stage/proposal.pdf] (March 22, 2009).

Gordon, A. S. (2006). Fourth Frame Forums: Interactive Comics for Collaborative Learning. In *Proceedings of the 14th Annual ACM International Conference on Multimedia*, pages 69–72, New York, NY, USA. ACM.

Grammenos, D., Mourouzis, A., and Stephanidis, C. (2006). Virtual Prints: Augmenting Virtual Environments with Interactive Personal Marks. *International Journal of Human-Computer Studies*, 64(3):221–239.

Gratz, J. M. (1981). The Origin of Spacewar. *Creative Computing*. [http://www.wheels.org/spacewar/creative/SpacewarOrigin.html] (March 22, 2009).

Greitzer, F. L., Kuchar, O. A., and Huston, K. (2007). Cognitive Science Implications for Enhancing Training Effectiveness in a Serious Gaming Context. *Journal of Educational Resources in Computing*, 7(3):2–16.

Grubert, J. and Habener, T. (2006/2007). Der virtuelle Tatort. Term paper. Otto-von-Guericke University, Department of Simulation and Graphics and University of Applied Sciences Magdeburg/Stendal, Industrial Design Institute, Magdeburg, Germany.

Grzeschik, R. (2006). Der virtuelle Werkzeugkasten. Internship report. Internship at Otto-von-Guericke University, Department of Simulation and Graphics, Magdeburg, Germany.

Haller, M., Holm, R., Priglinger, M., Volkert, J., and Wagner, R. (2000). Components for a Virtual Environment. Workshop on Guiding Users through Interactive Experiences: Usability Centred Design and Evaluation of Virtual 3D Environments. Paderborn.

Halper, N. and Masuch, M. (2003). Action Summary for Computer Games — Extracting and Capturing Action for Spectator Modes and Summaries. In *Proceedings of the 2nd International Conference on Application and Development of Computer Games*, pages 124–132, Hong Kong, China.

Harel, I. and Papert, S., editors (1991). *Constructionism (Cognition and Computing)*, chapter “Situating Constructionism”. Alex Publishing Corporation, Norwood, NJ, USA.

Harteveld, C. and Bidarra, R. (2007). Learning with Games in a Professional Environment: A Case Study of a Serious Game about Levee Inspection. In *Proceedings of Learning with Games 2007*, pages 555–562.

Hegner, M. (2003). Methoden zur Evaluation von Software. IZ-Arbeitsbericht 29, GESIS — InformationsZenrum Sozialwissenschaften, Bonn, Germany.

Hewicker, A. (2007). Darstellung von Fingerabdrücken. Term paper. Otto-von-Guericke University, Department of Simulation and Graphics, Magdeburg, Germany.

Hintze, A., Schumann, M., and Stuering, S. (2001). Distributed Virtual Training Applications for Education of Maintenance and Service Personnel. In *Proceedings of the 2001 European Simulation Interoperability Workshop*.

Hoobler, N., Humphreys, G., and Agrawala, M. (2004). Visualizing Competitive Behaviors in Multi-User Virtual Environments. In *Proceedings of the 15th IEEE Conference on Visualization 2004*, pages 163–170, Los Alamitos, CA, USA. IEEE Computer Society.

Howard, T., Hubbold, R., and Murta, A. (1999). MAVERICK: A Virtual Reality System for Research and Teaching. In *Graphics and Visualization Education 1999 (GVE99 Workshop)*.

Howard, T. L. J., Murta, A. D., and Gibson, S. (2000). Virtual Environments for Scene of Crime Reconstruction and Analysis. In *Proceedings of SPIE: Visual Data Exploration and Analysis VII*, volume 3960, pages 41–48, Bellingham WA, USA. Society of Photo-Optical Instrumentation Engineers.

Ingalls, D., Kaehler, T., Maloney, J., Wallace, S., and Kay, A. (1997). Back to the Future: The Story of Squeak, a Practical Smalltalk Written in Itself. In *Proceedings of the 12th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA '97)*, pages 318–326, New York, NY, USA. ACM.

Isenberg, T. and Brennecke, A. (2006). G-Strokes: A Concept for Simplifying Line Stylization. *Computer & Graphics*, 30(5):754–766. Elsevier Ltd.

Isenberg, T., Brennecke, A., Sousa, M. C., and Carpendale, S. (2005). Beyond Pixels: Illustration with Vector Graphics. Technical Report 2005-804-35, Department of Computer Science, University of Calgary, Canada.

Jenkins, H., Klopfer, E., Squire, K., and Tan, P. (2003). Entering the Education Arcade. *Computers in Entertainment (CIE)*, 1(1):17–17.

Jenkins, M., Browne, T., and Walker, R. (2005). VLE Surveys — A Longitudinal Perspective between March 2001, March 2003, and March 2005 for higher Education in the United Kingdom. Longitudinal Survey VLE Survey 2005, Universities and Colleges Information Systems Association.

Jool, J. (2005). *Half-Real : Video Games between Real Rules and Fictional Worlds*. The MIT Press, Cambridge, MA, USA.

Kansanen, P. and Meri, M. (1999). Didactic Relation in the Teaching-Studying-Learning Process. *Didaktik/Fachdidaktik as Science(-s) of the Teaching Profession*, 2(1):107–116.

Kay, A. (2005). Squeak Etoys, Children & Learning. VPRI Research Note RN-2005-001, Viewpoints Research Institute, Glendale, CA, USA.

Kay, A. C. (1993). The Early History of Smalltalk. *ACM SIGPLAN Notices*, 28(3):69–95.

Kearsley, G. (1982). Authoring Systems in Computer Based Education. *Communications of the ACM*, 25(7):429–437.

Kelly, H. (2005). Games, Cookies, and the Future of Education. *Issues in Science and Technology*, Summer. [http://www.issues.org/21.4/kelly.html] (March 22, 2009).

Kelly, H., Howell, K., Glinert, E., Holding, L., Swain, C., Burrowbridge, A., and Roper, M. (2007). How to Build Serious Games. *Communications of the ACM*, 50(7):44–49. Special Issue: Creating a Science of Games.

Korte, L., Anderson, S., Pain, H., and Good, J. (2007). Learning by Game-Building: A Novel Approach to Theoretical Computer Science Education. In *Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '07)*, pages 53–57, New York, NY, USA. ACM Press.

Korten, M., Mainka, M., and Schumacher, H. (2002). Game Based Interfaces. *Computer Graphik Topics — Reports on Computer Graphics*, 15:11–13.

Kosara, R., Healey, C. G., Interrante, V., Laidlaw, D. H., and Ware, C. (2003). User Studies: Why, How, and When? *Computer Graphics and Applications*, 23(4):20–25.

Kurlander, D., Skelly, T., and Salesin, D. (1996). Comic Chat. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pages 225–236, New York, NY, USA. ACM.

Lefrancois, G. R. (2006). *Psychologie des Lernens*. Springer-Verlag, Berlin, Germany, 3rd edition.

Lin, C.-C., Li, Y.-C., Bai, Y.-M., Chen, J.-Y., Hsu, C.-Y., Wang, C.-H., Chiu, H.-W., and Wan, H.-T. (2005). The Evaluation of Game-based E-learning for Medical Education: a Preliminary Survey. *AMIA Annual Symposium Proceedings*, page 1032.

Linzmayer, O. W. (1999). *Apple Confidential — The Real Story of Apple Computer, Inc.* No Star Press, San Francisco, CA, USA.

Malone, T. W. (1982). Heuristics for Designing Enjoyable User Interfaces: Lessons from Computer Games. In *Proceedings of the 1982 Conference on Human factors in Computing Systems*, pages 63–68, New York, NY, USA. ACM.

Marshall, G. (1998). Sociology of Training. In A Dictionary of Sociology off HighBeam Encyclopedia. [http://www.encyclopedia.com/] (March 22, 2009).

Martens, A., Diener, H., and Malo, S. (2008). Game-Based Learning with Computers — Learning, Simulations, and Games. *Transactions on Edutainment*, 5080:172–190.

Masuch, M., Schlechtweg, S., and Schulz, R. (1999). Speedlines: Depicting Motion in Motionless Pictures. In *SIGGRAPH 99 Conference Abstracts and Applications*.

Maxwell, B. A., Ward, N., and Heckel, F. (2004). Game-Based Design of Human-Robot Interfaces for Urban Search and Rescue. In *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems*. ACM Press.

McCloud, S. (1994). *Understanding Comics - The Invisible Art*. HarperPerennial, New York.

McLuhan, M. (2001). *Understanding Media — The Extension of Man*, volume 2. Taylor and Francis.

Michael, D. and Chen, S. (2006). *Serious Games. Games that Educate, Train, and Inform*. Thomson Course Technology PTR, Boston, 1st edition.

Mitchell, D. (1995). From MUDs to Virtual Worlds. Technical report, Microsoft Virtual Worlds Group.

Müller, M.-L. (2006). Benutzerschnittstelle für die Tatortsicherung. Diploma thesis, Otto-von-Guericke University of Magdeburg, Department of Simulation and Graphics, Magdeburg, Germany.

Nelson, M. J. and Mateas, M. (2008). An Interactive Game-Design Assistant. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '08)*, pages 90–98, New York, NY, USA. ACM.

Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Nielsen, J. (1994). *Usability Inspection Methods*, chapter "Heuristic Evaluation", pages 25–62. John Wiley & Sons, Inc., New York, NY, USA.

Nielsen, J. (n.d.). How to Conduct a Heuristic Evaluation. [http://www.useit.com/papers/heuristic/heuristic_evaluation.html] (March 22, 2009).

Nielsen, J. and Mack, R. L., editors (1994). *Usability Inspection Methods*. John Wiley & Sons, Inc., New York, NY, USA.

Nielsen, J. and Molich, R. (1990). Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*, pages 249–256, New York, NY, USA. ACM.

Nienhaus, M. and Döllner, J. (2005). Depicting Dynamics Using Principles of Visual Art and Narrations. *IEEE Computer Graphics and Applications*, 25(3):40–51.

NMC (2007). Spring 2007 Survey — Educators in Second Life. Technical report, The New Media Consortium. [http://www.nmc.org/pdf/2007-sl-survey-summary.pdf] (March 22, 2009).

NPD Group Inc. (2008). 2007 U.S. Video Game and PC Game Sales Exceed $18.8 Billion Marking Third Consecutive Year of Record-Breaking Sales. Press release. [http://www.npd.com/press/releases/press_080131b.html] (March 22, 2009).

NPD Group Inc. (2009). 2008 Video Game Software Sales Across Top Global Markets Experience Double-Digit Growth. Press release. [http://www.npd.com/press/releases/press_090202.html] (March 22, 2009).

Olwal, A. and Feiner, S. (2003). The Flexible Pointer: An Interaction Technique for Selection in Augmented and Virtual Reality. Conference Supplement of UIST '03 (ACM Symposium on User Interface Software and Technology). pp. 81–82.

Panzer, C. (2007). Die virtuelle SLR-Kamera für die Tatortfotografie. Diploma thesis, Otto-von-Guericke University of Magdeburg, Department of Simulation and Graphics, Magdeburg, Germany.

Papert, S. (1980). *MindStorms: Children, Computers and Powerful Ideas*. Basic Books, Inc., New York, NY, USA.

Papert, S. (1987). Microworlds: Transforming Education. *Artificial Intelligence and Education; Vol. 1: Learning Environments and Tutoring Systems*, pages 79–94.

Papert, S. (1999). What is Logo? Who Needs It? *Logo Philosophy and Implementation*, pages IV–XVII. Foreword.

Paras, B. and Bizzocchi, J. (2005). Game, Motivation, and Effective Learning: An Integrated Model for Educational Game Design. In *The International DiGRA Conference*, Vancouver, Canada.

Patil, S. and Howell, K. (2007). My Learning Assistant: Question-Asking and -Answering in Synthetic Game Environments. *interactions*, 14(1):43–45.

Pausch, R., Gold, R., Skelly, T., and Thiel, D. (1994). What HCI Designers Can Learn From Video Game Designers. In *CHI '94: Conference Companion on Human Factors in Computing Systems*, pages 177–178, New York, NY, USA. ACM.

PDV (1985). *Polizeidienstverordnung 100*. Boorberg Verlag, Stuttgart, Germany.

Petterson, L. W., Jensen, N., and Seipel, S. (2003). A Virtual Laboratory for Computer Graphics Education. In *Eurographics 2003*, pages 1–6. Eurographics Association.

Piaget, J. (2003). *Meine Theorie der geistigen Entwicklung*. Beltz Verlag, Weinheim.

Pinelle, D., Wong, N., and Stach, T. (2008). Heuristic Evaluation for Games: Usability Principles for Video Game Design. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '08)*, pages 1453–1462, New York, NY, USA. ACM.

Pivec, M., Dziabenko, O., and Schinnerl, I. (2003). Aspects of Game-Based Learning. In *The 3rd International Conference on Knowledge Management (I-KNOW'03)*, pages 217–224, Graz, Austria. Graz and Journal of Universal Computer Science (JUCS).

Prange, K. (1983). *Bauformen des Unterrichts*. Klinkhardt, Bad Heilbronn/Obb, Germany.

Prensky, M. (2001). *Digital Game-Based Learning*. McGraw-Hill Companies, New York, 1st edition.

Prensky, M. (2003). Digital Based Learning. *Computers in Entertainment (CIE)*, 1(1):21–25. COLUMN: Book Synopsis — Books et. al.

Quinn, C. N. (1996). Pragmatic Evaluation: Lessons from Usability. The 13th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (Ascilite '96). [http://www.ascilite.org.au/conferences/adelaide96/papers/18.html] (March 22, 2009).

Quinn, C. N. (2005). *Engaging Learning — Designing e-Learning Simulation Games*. Pfeiffer, San Francisco, CA, USA.

Raybourn, E. M. and Bos, N. (2005). Design and Evaluation Challenges of Serious Games. In *Conference on Human Factors in Computing Systems, CHI 2005 Extended Abstracts on Human Factors in Computing Systems*, pages 2049–2050, New York, NY, USA. ACM Press. SESSION: Special Interest Groups (SIG).

Rehbein, F., Kleimann, M., and Mößle, T. (2009). Computerspielabhängigkeit im Kindes- und Jugendalter: Empirische Befunde zu Ursachen, Diagnostik und Komorbiditäten unter besonderer Berücksichtigung spielimmanenter Abhängigkeitsmerkmale. KFN-Forschungsbericht 108, Kriminologisches Forschungsinstitut Niedersachsen e.V. (KFN), Hannover, Germany. [http://www.kfn.de/home/Computerspielabhaengigkeit_im_Kindes-_und_Jugendalter.htm] (March 22, 2009).

Reinemann, K. (2006). Kriminalistisch relevante Spuren im virtuellen Raum. Diploma thesis, Otto-von-Guericke University of Magdeburg, Department of Simulation and Graphics, Magdeburg, Germany.

Reinemann, K. and Hähnel, P. (2005/2006). Modeling of Virtual 3D Crime Scenes. Student assignment. Otto-von-Guericke University, Department of Simulation and Graphics.

Reinke, N. and Seidl, T. (2006/2007). Der virtuelle Tatort. Term paper. Otto-von-Guericke University, Department of Simulation and Graphics and University of Applied Sciences Magdeburg/Stendal, Industrial Design Institute, Magdeburg, Germany.

Rieber, L. P. (1996). Seriously Considering Play: Designing Interactive Learning Environments Based on the Blending of MicroWorlds, Simulations, and Games. *Educational Technology Research & Development*, 44(2):43–58.

Robertson, J. and Good, J. (2005). Story Creation in Virtual Game Worlds. *Commun. ACM*, 48(1):61–65.

Ryan, R. M. and Deci, E. L. (2000). Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. *American Psychologist*, 55(1):68–78.

Ryan, T. (1999). Beginning Level Design, Part 1: Level Design Theory. *Gamasutra*. [http://www.gamasutra.com/features/19990416/level_design_01.htm] (March 22, 2009).

Saffer, D. (2006). *Designing for Interaction: Creating Smart Applications and Clever Devices*. Peachpit Press, Berkeley, CA, USA.

Salen, K. and Zimmerman, E. (2004). *Rules of Play: Game Design Fundamentals*. The MIT Press, London, UK.

Schmidgall, A. J. (2006). Effect of Violent Video Games on Aggression. Manuscript, Department of Psychology, Missouri Western State University. [http://clearinghouse.missouriwestern.edu/manuscripts/810.php] (March 22, 2009).

Schneider, D. K. and Godard, R. (1996). Virtual Environments for Education, Research and Life. Position Paper, 5th International World Wide Web Conference, Workshop W5: Virtual Environments and the WWW.

Schulze, M. and Maertins, R. (2006). Charaktermodellierung & -animation. Term paper. Otto-von-Guericke University, Department of Simulation and Graphics, Magdeburg, Germany.

Schumann, H. and Müller, W. (2000). *Visualisierung: Grundlagen und allgemeine Methoden*. Springer, Dordrecht, The Netherlands, 1 edition.

Se, S. and Jasiobedzki, P. (2005). Instant Scene Modeler for Crime Scene Reconstruction. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05) - Workshops*, page 123, Los Alamitos, CA, USA. IEEE Computer Society.

Shaffer, D. W., Squire, K. R., Halverson, R., and Gee, J. P. (2005). Video Games and the Future of Learning. WCER Working Paper 2005–4, University of Wisconsin-Madison, WI, USA.

Sickel, R. (2006). Protokollierung und Visualisierung von Nutzerinteraktionen in Serious Games. Diploma thesis, Otto-von-Guericke University of Magdeburg, Department of Simulation and Graphics, Magdeburg, Germany.

Silius, K. and Tervakari, A.-M. (2003). The Usefulness of Web-Based Learning Environments — The Evaluation Tool into the Portal of Finnish Virtual University. In *International Conference on Network Universities and e-Learning*, page n.a., Valencia, Spain. Proceedings of mENU2003 (CD-ROM).

Skinner, B. (1974). *About Behaviorism*. Knopf, New York, NY, USA.

Smith, D. A., Raab, A., Reed, D. P., and Kay, A. (2004). Croquet: A Menagerie of New User Interfaces. In *Proceedings of the 2nd International Conference on Creating, Connecting and Collaborating through Computing (C5 '04)*, pages 4–11, Washington, DC, USA. IEEE Computer Society.

Sousa, M. C. (2003). Scientific Illustration Part 1: Traditional Techniques and NPR Approaches. In Sousa, M. C., editor, *Theory and Practice of Non-Photorealistic Graphics: Algorithms, Methods, and Production Systems*, volume 10 of *ACM SIGGRAPH 2003 Course Notes*, chapter 5. ACM SIGGRAPH, New York.

Spengler, M. (2006). Interaktive Abfrage von Rechtsgrundlagen bei der Tatortarbeit. Diploma thesis, Otto-von-Guericke University of Magdeburg, Department of Simulation and Graphics, Magdeburg, Germany.

Squire, K. and Barab, S. (2004). Replaying History: Engaging Urban Underserved Students in Learning World History through Computer Simulation Games. In *Proceedings of the 6th International Conference on Learning Sciences (ICLS '04)*, pages 505–512. International Society of the Learning Sciences.

Squire, K. D. (2004). *Replaying History: Learning World History Through Playing*. Dissertation, School of Education, Instructional Systems Technology Department, Indiana University, Indiana, USA.

Squires, D. and Preece, J. (1999). Predicting Quality in Educational Software: Evaluating for Learning, Usability, and the Synergy between them. *Interacting with Computers*, 11(5):467–483.

Ssemugabi, S. and de Villiers, R. (2007). A Comparative Study of two Usability Evaluation Methods Using a Web-Based e-Learning Application. In *Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries (SAICSIT '07)*, pages 132–142, New York, NY, USA. ACM Press.

Strothotte, T. and Schlechtweg, S. (2002). *Non-Photorealistic Computer Graphics. Modeling, Rendering, and Animation*. Morgan Kaufmann Publishers, San Francisco, CA, USA.

Susi, T., Johannesson, M., and Backlund, P. (2007). Serious Games — An Overview. Technical Report HS-IKI-TR-07-001, School of Humanities and Informatics, University of Skövde, Sweden.

Swing, E. L. and Anderson, C. A. (2008). *Children's Learning in a Digital World*, chapter "How and what do video games teach?", pages 64–84. Blackwell, Oxford, UK.

Szilas, N., Barles, J., and Kavakli, M. (2007). An Implementation of Real-Time 3D Interactive Drama. *Computers and Entertainment*, 5(1):5.

Tavanti, M. and MatsLind (2001). 2D vs 3D, Implications on Spatial Memory. *IEEE Symposium on Information Visualization, 2001. (INFOVIS 2001)*, pages 139–145.

Taxen, G. and Naeve, A. (2001). CyberMath: Exploring Open Issues in VR-Based Learning. In *ACM SIGGRAPH 2001 Conference Abstracts and Applications*, pages 49–51, New York, NY, USA. ACM Press. SIGGRAPH 2001 Educators Program.

Teyseyre, A. R. and Campo, M. R. (2009). An Overview of 3D Software Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):87–105.

The Columbia Encyclopedia (2007). In HighBeam Encyclopedia. [http://www.encyclopedia.com/] (March 22, 2009).

The Oxford Pocket Dictionary of Current English (2008). In HighBeam Encyclopedia. [http://www.encyclopedia.com/] (March 22, 2009).

Tominski, C. (2006). *Event-Based Visualization for User-Centered Visual Analysis*. Dissertation, University of Rostock, Rostock, Germany.

van Eck, R. (2006). Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless. *EDUCAUSE Review*, 41(2):16–30.

Viola, I. and Sousa, M. C. (2006). Focus of attention+context and Smart Visibility in Visualization. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 5, New York, NY, USA. ACM.

Wagner, F., Schmuki, R., Wagner, T., and Wolstenholme, P. (2006). *Modeling Software with Finite State Machines: A Practical Approach*. CCR Press (Tayler & Francis Group), Boca Raton, FL, USA.

Ware, C. (2001). Designing with a 2 1/2D Attitude. *Information Design Journal*, 10(3):255–262.

Warmerdam, J., Knepflé, M., Bekebrede, G., Mayer, I., and Bidarra, R. (2007). The Serious Game Simport: Overcoming Technical Hurdles in Educational Gaming. In *Proceedings of Learning with Games 2007*, pages 307–314.

Wolley, D. R. (1994). PLATO: The Emergence of Online Community. *Matrix News*. [http://thinkofit.com/plato/dwplato.htm] (March 22, 2009).

Wong, W. L., Shen, C., Nocera, L., Carriazo, E., Tang, F., Bugga, S., Narayanan, H., Wang, H., and Ritterfeld, U. (2007). Serious Video Game Effectiveness. In *Proceedings of the*

International Conference on Advances in Computer Entertainment Technology (ACE'07), pages 49–55, New York, NY, USA. ACM.

Zazelenchuk, T. (2006). Heuristic Evaluation and its Alternatives. *Userfocus — User Experience Research, Design & Training*. [http://www.userfocus.co.uk/articles/heuristics.html] (March 22, 2009).

Zyda, M., Mayberry, A., Wardynski, C., Shilling, R., and Davis, M. (2003). The MOVES Institute's America's Army Operations Game. In *Proceedings of the 2003 Symposium on Interactive 3D Graphics (I3D '03)*, pages 219–220, New York, NY, USA. ACM.

Appendix A

Supervised Student Assignments

Diploma Theses

Marie-Luise Müller: *Benutzerschnittstelle für die Tatortsicherung.*
Otto-von-Guericke University Magdeburg, 2006/09.

Christian Panzer: *Die virtuelle SLR-Kamera für die Tatortfotografie.*
Otto-von-Guericke University Magdeburg, 2007/07.

Kerstin Reinemann: *Kriminalistisch relevante Spuren im virtuellen Raum.*
Otto-von-Guericke University Magdeburg, 2006/11.

René Sickel: *Protokollierung und Visualisierung von Nutzerinteraktionen in Serious Games.*
Otto-von-Guericke University Magdeburg, 2007/02.

Michael Spengler: *Interaktive Abfrage von Rechtsgrundlagen bei der Tatortarbeit.*
Otto-von-Guericke University Magdeburg, 2006/09.

Internship Report

Ramona Grzeschik: *Der virtuelle Werkzeugkasten.*
Otto-von-Guericke University Magdeburg, 2006/02.

Term Papers

Tom Brosch: *Blutspurenmodellierung.*
Otto-von-Guericke University Magdeburg, 2006/04.

Alexander Hewicker: *Darstellung von Fingerabdrücken.*
Otto-von-Guericke University Magdeburg, 2007/06.

Jens Grubert and Thomas Habener: *Der virtuelle Tatort.*
Collaboration project between University of Applied-Sciences Magdeburg-Stendal & Otto-von-Guericke University Magdeburg, 2006/2007.

Nicole Reinke and Thomas Seidl: *Der virtuelle Tatort.*
Collaboration project between University of Applied-Sciences Magdeburg-Stendal & Otto-von-Guericke University Magdeburg, 2006/2007.

Maik Schulze and Robert Maertins: *Charaktermodellierung & -animation.*
Otto-von-Guericke University Magdeburg, 2006/04.

Appendix B

Heuristics

The heuristics used for the system evaluation are presented in the following tables (cf. Sec. 5.4).

Heuristic	Description
Clear goals and objectives	The learner should understand what is to be accomplished and what is to be gained from use.
Context meaningful to domain and learner	The activity should be situated in practice and engaging to the learner.
Content clearly and multiply represented, and multiply navigable	The message should be unambiguous and support different learner preferences, and allow the learner to find relevant information while engaged in an activity.
Activities scaffolded	Learner activities need to be supported to allow working within competence yet on meaningful chunks of knowledge.
Elicit learner understandings	Learners need to articulate their conceptual understandings as the basis for feedback.
Formative evaluation	Learners need constructive feedback on their endeavours.
Performance should be "criteria-referenced"	The outcomes should be clear and measurable; competency-based evaluation should be a goal.
Support for transference and acquiring "self-learning" skills	The environment should support transference of the skills beyond the learning environment, and facilitate the learner becoming able to self-improve.

Table B.1.: The table lists the heuristics developed by Quinn [1996] to inspect and evaluate educational software.

Heuristic	Description
Establishment of context	The photographs, documents and other materials related to the simulated schools create a sense of immersion in a simulated reality.
Relevance to professional practice	The problem scenarios and included tasks are realistic and relevant to the professional practice of teachers.
Representation of professional responses to issues	The sample solutions represent a realistic range of teacher responses to the issues and challenge users to consider alternative approaches.
Relevance of reference materials	The reference materials included in the package are relevant to the problem scenarios and are at a level appropriate to the users.
Presentation of video resources	The video clips of teacher interviews and class activities are relevant and readily accessible to the user.
Assistance is supportive rather than prescriptive	The contextual help supports the user in locating relevant resources and dealing with the scenarios without restricting the scope of individual responses.
Materials are engaging	The presentation style and content of the software encourages a user to continue working through the scenarios.
Presentation of resources	The software presents useful resources for teacher professional development in an interesting and accessible manner.
Overall effectiveness of materials	The materials are likely to be effective in increasing teachers' confidence and capacity for integrating information technology into teaching and learning.

Table B.2.: The table lists the heuristics developed by Albion [1999] to inspect and evaluate the content of educational software.

Heuristic	Description
Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Match between system and real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
Error prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
Flexibility and efficiency of use	Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
Aesthetic and minimalistic design	Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Table B.3.: The table lists the heuristics developed by Nielsen [1994] to inspect and evaluate user interface design. See also [http://www.useit.com/papers/heuristic/heuristic_list.html] (March 22, 2009).

Appendix C

Questionnaire

The questionnaire that was used for the system evaluation is presented in this appendix (cf. Sec. 5.4).

Fragebogen

zur Verwendung des
virtuellen Tatorts als Lehr- und Lernmaterial

Dipl.-Ing. Angela Brennecke
Universität Rostock

30.06.2008

Sehr geehrte Damen, sehr geehrte Herren,

im Rahmen des Kollaborationsprojekts „Der virtuelle Tatort“ zwischen der Fachhochschule Polizei Sachsen-Anhalt und der Otto-von-Guericke Universität Magdeburg ist die prototypische Anwendung OpenCrimeScene zum virtuellen Tatortsicherungs-training entstanden. Das Programm ermöglicht in seinem aktuellen Zustand zwei Dinge: Ein virtueller Täter kann Finger-, Schuh-, und Blutspuren in einem virtuellen Raum platzieren und Gegenstände entwenden. Der entstandene virtuelle Tatort kann dann von einem virtuellen Polizisten betreten werden. Dieser kann mit einer virtuellen SLR-Kamera die gefundenen Spuren fotografieren.

Dieser Fragebogen soll dazu beitragen, die aktuelle Umsetzung des Prototypen auszuwerten. Somit kann eine für die Polizeihochschule angemessene Weiterentwicklung erheblich erleichtert werden. Hierzu möchte ich Sie bitten, eine Reihe von Fragen zu beantworten. Die Fragen sind in drei Kategorien unterteilt: 1. Benutzungsschnittstelle, 2. Mehrwert für die Lehre und das Studium und 3. Neue Konzepte. Kategorie 1 soll darüber Aufschluss geben, wie handhabbar das Programm ist, ob die grafische Darstellung eindeutig und leicht verständlich ist, etc. Kategorie 2 soll dazu beitragen, festzustellen, inwiefern Sie einen Mehrwert für Lehre und Studium mit dem Programm einschätzen und wie Sie die spielbasierte Umsetzung bewerten. Letztendlich soll Kategorie 3 darüber Aufschluss geben, wie die neuen Konzepte von Ihnen eingeschätzt werden.

Die Fragen der drei Hauptkategorien basieren auf grundsätzlichen Richtlinien (Heuristiken) aus der Informatik, die bei der Entwicklung von Software-Anwendungen verwendet werden, um eine gute Einsatzfähigkeit und Nutzbarkeit von Software-Anwendungen zu gewährleisten. Ob der OpenCrimeScene-Prototyp bereits „auf dem richtigen Weg“ ist, soll anhand der auf diesen Richtlinien basierenden Fragen ausgewertet werden.

Zusätzlich zu den drei Hauptkategorien möchte ich Sie bitten, noch ein paar allgemeine Angaben über Ihre bisherigen Kenntnisse bzgl. Computeranwendung zu machen. Auch werde ich abschließend noch ein paar wenige Fragen über mögliche Weiterentwicklungen des Programms an Sie richten.

Für die Auswertung der Fragebögen soll die in der Informatik gängige Methode der „Heuristischen Evaluation“ vorgenommen werden. Diese sieht vor, dass ca. 3 Expertengruppen mit jeweils 3-5 Experten an der Befragung teilnehmen. Es wird dabei davon ausgegangen, dass die Experten sich in die Rolle der Endnutzer hineinversetzen und deren Verhalten abschätzen können. Für diese Befragung sind die Expertengruppen Lehrer, Schüler und Informatiker von Interesse, mit je 3-5 Teilnehmern. Diese Zahl ist ausreichend für eine erste Abschätzung.

Die beiliegende CD-ROM enthält ein Demoprogramm und ein Video von OpenCrimeScene. Für die Evaluierung soll entweder das Video oder das Video und das Demoprogramm verwendet werden. Die Expertengruppen sollen sich hierfür in zwei möglichst gleichgroße Gruppen aufteilen. Gruppe 1 soll sich nur das Video anschauen und dann mit der Beantwortung der Fragebögen beginnen. Gruppe 2 soll sich das Video anschauen und das Gesehene dann mit dem Demoprogramm nachspielen. Danach soll die Beantwortung der Fragen durchgeführt werden.

Um sich das Video anzusehen, können Sie das beigefügte Programm „vlc.exe“ auf Ihrem PC installieren, sofern das Video nicht mit Ihrem vorinstallierten Player laufen sollte. Um mit dem Demoprogramm zu arbeiten, müssen Sie den Ordner „OpenCrimeScene“ in ein Verzeichnis auf Ihrem PC kopieren und dann den Ordner „OpenCrimeScene“ öffnen. In diesem Ordner befindet sich die Datei „LIESMICH.txt“, die Sie über die notwendigen Tasten- und Mauskombinationen aufklärt. Zum Starten des Programms müssen Sie die Datei „OCS_Start.bat“ doppelt anklicken. Sie müssen das Video nicht komplett nachvollziehen. Jedoch wäre es für den Fragebogen wichtig, wenn Sie im virtuellen Tatort als virtueller Täter Spuren platzieren oder als virtueller Polizist Spuren fotografieren. Falls das Programm aus technischen Gründen nicht laufen sollte, geben Sie diese bitte an und fahren Sie nach Betrachtung des Videos mit dem Beantworten der Fragen fort.

Die Fragen betreffen Ihren Eindruck von OpenCrimeScene. Bewerten Sie die Fragen bitte auf einer Skala von sehr wenig (--) bis (++) sehr gut. Wenn Sie sich nicht sicher sind oder keine Angaben machen können, kreuzen Sie bitte das Feld „k.A.“ (keine Angaben) an. Dies ist durchaus möglich, bspw. wenn Sie zur Videogruppe gehören und keine Angaben zu Programminteraktionen machen können. Benutzen Sie für Kommentare bitte die Rückseite des Formulars und geben Sie immer die Nummer der jeweiligen Frage mit an.

Für die Beantwortung der Fragen können Sie sich das Video gerne mehrmals ansehen oder das Programm verwenden. Zur weiteren Unterstützung finden Sie im Anschluss an die allgemeinen Fragen eine Reihe an Screenshots aus dem Programm.

Die Experten sollen die Beantwortung der Fragen getrennt voneinander vornehmen und vor Beendigung der Beantwortung keine Informationen austauschen, um eine unabhängige Auswertung sicherstellen zu können. Nachdem die Experten die Fragebögen ausgefüllt haben, ist eine Diskussion und ein Kommentar erwünscht. Alle Angaben werden selbstverständlich anonym behandelt. Die Dauer der Evaluierung wird auf 30-60 Min. geschätzt.

Für Ihre Zeit und Unterstützung möchte ich mich schon jetzt ganz herzlich bedanken. Die Ergebnisse der Auswertung teile ich Ihnen gerne mit. Auch wird die gesamte Evaluierung einen Teil meiner Dissertation ausmachen. Diese stelle ich Ihnen nach Beendigung gerne zur Verfügung.

Mit freundlichen Grüßen,

Angela Brennecke

Allgemeine Fragen

Wie alt sind Sie? ___ Jahre

Geschlecht? weiblich ☐ männlich ☐

Zu welcher Expertengruppe gehören Sie? Dozent(in) FH Polizei ☐ Student(in) FH Polizei ☐ Informatiker(in) ☐

Folgende Werteskala ist anzuwenden: sehr wenig (--), wenig (-), normal (o), gut (+) und sehr gut (++).

	--	-	o	+	++
Wie schätzen Sie Ihre Erfahrung im Umgang mit dem PC ein?	☐	☐	☐	☐	☐
Wie schätzen Sie Ihre Erfahrung mit Computerspielen ein?	☐	☐	☐	☐	☐

Haben Sie bereits mit PC-Anwendungen im Unterricht gearbeitet? ja ☐ nein ☐ k.A. ☐

Wenn ja, welche Anwendungen haben Sie verwendet? ______________________

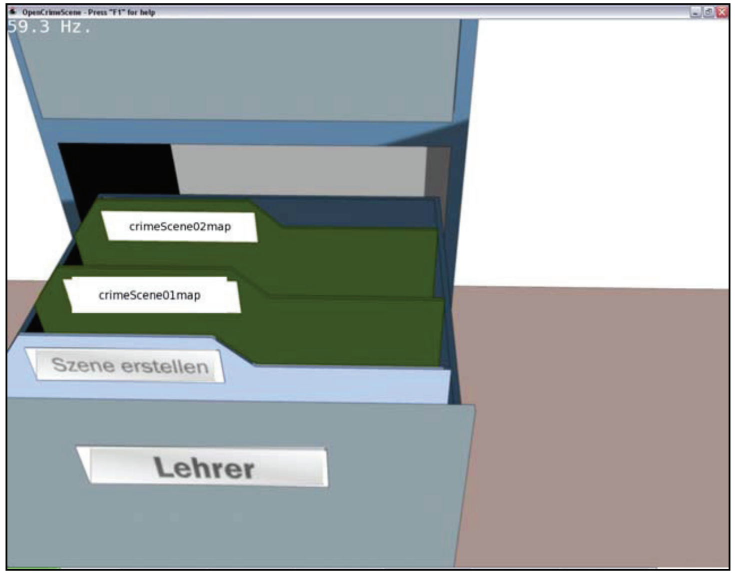
Haben Sie bereits mit Spielen (allgemein, PC, …) im Unterricht gearbeitet? ja ☐ nein ☐ k.A. ☐

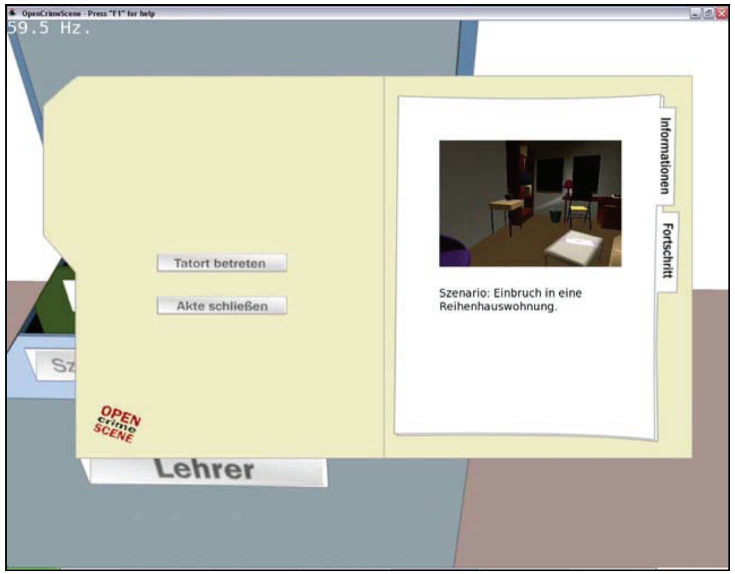
Wenn ja, welche Spiele haben Sie verwendet? ______________________

Halten Sie spielbasierte Computeranwendungen für das Training für lernförderlich? ja ☐ nein ☐ k.A. ☐

Würden Sie spielbasierte Computeranwendungen für das Training versuchsweise im Unterricht verwenden wollen? ja ☐ nein ☐ k.A. ☐

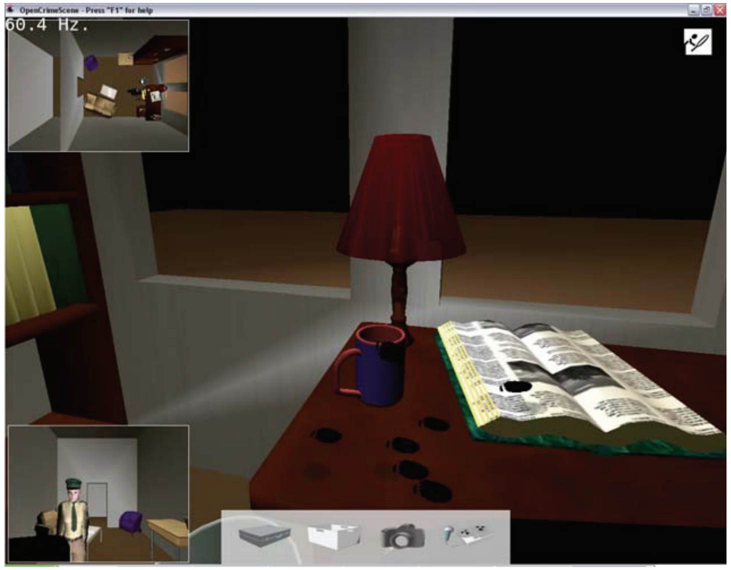
Das Hauptmenü

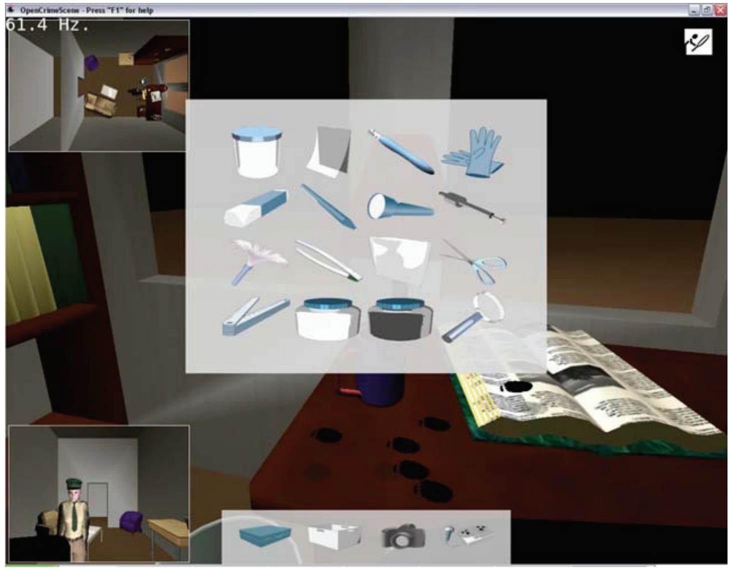


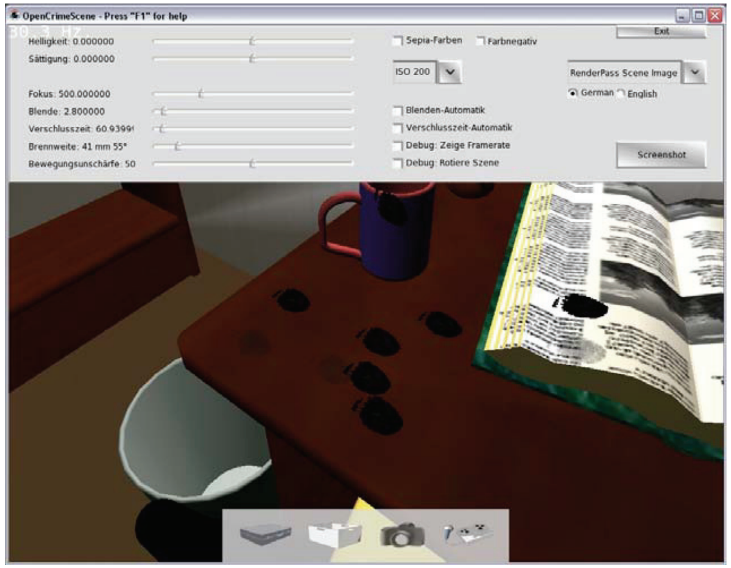


Das Lehrer-/Tätermenü

Das Studenten-/Polizistenmenü







Fragebogen Kategorie 1: Benutzungsschnittstelle und Programmaufbau

		--	-	o	+	++	k.A.
1.	War Ihnen während des Video-/Programmdurchlaufs klar, in welchem Programmteil Sie sich gerade befunden haben und worum es dabei ging?	☐	☐	☐	☐	☐	☐
2.	Waren die dargestellten Menünamen eindeutig und verständlich?	☐	☐	☐	☐	☐	☐
3.	Wie bewerten Sie die Freiheiten der Nutzerkontrolle (Navigation, Interaktion, Fehlerbehebung, Beenden des Programms, …)?	☐	☐	☐	☐	☐	☐
4.	Wie vertraut kommt Ihnen der allgemeine Programmaufbau im Vergleich zu Ihnen bekannten Computeranwendungen?	☐	☐	☐	☐	☐	☐
5.	Bewerten Sie, wie das Programm den Nutzer beim Vermeiden von Fehlern unterstützt? (bspw. wenn ein Objekt fälschlicherweise selektiert wurde)	☐	☐	☐	☐	☐	☐
6.	Sind die Interaktionsmöglichkeiten per Maus oder über das Menü leicht zu erinnern bzw. einfach zu erreichen?	☐	☐	☐	☐	☐	☐
7.	Schätzen Sie ein, ob Sie effizient mit dem Programm arbeiten könnten, bspw. um als virtueller Täter einen Verbrechen zu begehen.	☐	☐	☐	☐	☐	☐
8.	Halten Sie das grafische Design für ansprechend?	☐	☐	☐	☐	☐	☐
9.	Sind die Fehlermeldungen des Programms verständlich?	☐	☐	☐	☐	☐	☐
10.	Wie bewerten Sie die Dokumentation des Programms?	☐	☐	☐	☐	☐	☐

Fragebogen Kategorie 2: Mehrwert für Lehre und Studium

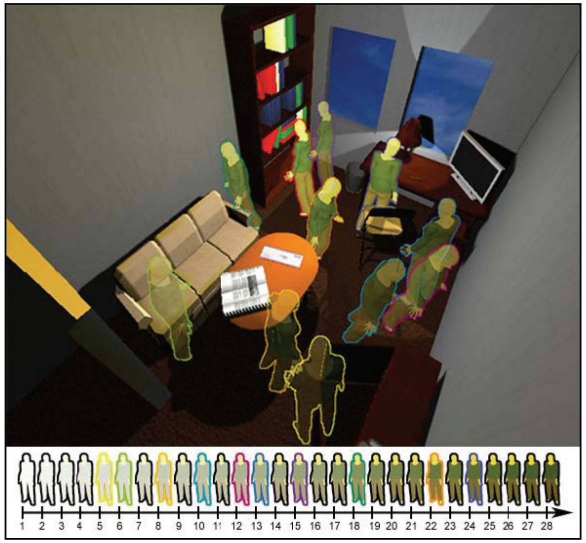
		--	-	o	+	++	k.A.
1.	Sind Ihnen Aufgabe und Ziel des jeweiligen Szenarios klar geworden?	☐	☐	☐	☐	☐	☐
2.	Konnten Sie eine Beziehung zwischen dem Programmkontext und dem realen Training herstellen?	☐	☐	☐	☐	☐	☐
3.	Wie gefällt es Ihnen, das Training als Spiel „Täter vs. Polizist“ zu gestalten?	☐	☐	☐	☐	☐	☐
4.	Kann das Trainingsspiel das Studium ansprechender und effektiver machen, bspw. um bereits Gelerntes zu vertiefen?	☐	☐	☐	☐	☐	☐
5.	Kann das Programm in den Lehrplan integriert werden? Bitte geben Sie dazu einen Kommentar ab, wie dies geschehen könnte.	☐	☐	☐	☐	☐	☐
6.	Würden Sie ein solches Trainingsspiel für das Studium verwenden wollen?	☐	☐	☐	☐	☐	☐

Fragebogen Kategorie 3: Neue Konzepte

		--	-	o	+	++	k.A.
1.	Wie bewerten Sie, dass der Dozent als aktiver Spieler agiert?	☐	☐	☐	☐	☐	☐
2.	Wie bewerten Sie, dass verschiedene Szenarien als Teil des Spiels erstellt werden können?	☐	☐	☐	☐	☐	☐
3.	Wie bewerten Sie, dass das Programm reale Trainingsmittel wie z.B. die SLR-Kamera zur Tatortsicherung zur Verfügung stellt?	☐	☐	☐	☐	☐	☐
4.	Wäre es für Sie vorteilhaft, wenn die Darstellung der Trainingsmittel exakt der Realität entsprechen würde?	☐	☐	☐	☐	☐	☐
5.	Würden Sie es für einen Mehrwert halten, wenn der Student in die Rolle des Täters schlüpfen könnte?	☐	☐	☐	☐	☐	☐

Bitte beantworten Sie abschließend noch zwei Fragen auf der nächsten Seite, die den weiteren Ablauf der Entwicklung betreffen.

Ausblickende Fragen



Der virtuelle Tatort soll als zusätzliches Lehr- und Lernmaterial zum Unterricht beitragen. Eine mögliche Integration in den Unterricht bestünde darin, Tatortszenarien also virtueller Täter zu entwerfen und die Studenten den Tatort sichern zu lassen. Die Abbildung links zeigt ein Protokoll über die Bewegungen des Studenten im virtuellen Raum. Die Transparenzwerte und farbigen Silhouetten geben über die jeweiligen Zeitpunkte Auskunft. Genauere Darstellungsformen des Geschehenen wie z.B. kurze Bildergeschichten, die zusätzlich Fehler oder angewendete Sicherungstechnik hervorheben, sollen vom Programm automatisch generiert werden können.

1. Könnten Sie sich vorstellen, solche Auswertungsprotokolle im Unterricht zu diskutieren? ja ☐ nein ☐ k.A. ☐
Bitte kommentieren Sie Ihre Meinung auf der Rückseite.
2. Ein SmartEnvironment ist ein realer Raum, der mit Computertechnik vernetzt ist. Ein zukünftiges Szenario könnte darin so aussehen, dass mittels einer speziellen Brille Informationen eingeblendet werden (bspw. so ähnlich wie in den Filmen Mission Impossible oder Minority Report). Könnten Sie sich vorstellen, dass so etwas für das Training förderlich ist? ja ☐ nein ☐ k.A. ☐

Falls Sie weitere Anmerkungen, Ideen o.a. für die Verwendung des virtuellen Tatorts haben, formulieren Sie diese bitte auf der Rückseite dieses Formulars. Damit wäre die Befragung am Ende angelangt.
Ich möchte Ihnen herzlich für Ihre Mitarbeit und Unterstützung danken!

Appendix D

Results of the System Evaluation

The results of the system evaluation are presented in the following tables (cf. Sec. 5.4). The tables below contain a sequence of bars that are structured as follows: teachers (dark gray bars), students (light gray bars), and developers (white bars). Values by female interviewees are underlined. For example, a rating by 3 female and 1 male interviewee would be represented as "$\underline{3}$/1." The expert groups consisted of 5 teachers, 6 students, and 4 developers. The average value ∅ denotes the arithmetic mean.

	General Questions					
	$Q1$			$Q2$		
++ (5)	2	1	3	1	1	3
+ (4)	2	$\underline{2}$/2	1		$\underline{1}$/2	
o (3)	1	$\underline{1}$			$\underline{1}$	
− (2)				4	$\underline{1}$	
−− (1)						1
n/a						
∅	4,2	4,0	4,8	2,6	3,7	4,0

Table D.1.: Results of the general questions $Q1$–$Q2$.

	General Questions											
	$Q3$			$Q4$			$Q5$			$Q6$		
yes	5	3/1	2	1			5	3/3	4	5	3/3	3
no		2		4	3/3	2						
n/a			2			2						1

Table D.2.: Results of the general questions $Q3$–$Q6$.

	Usability Principles														
	$Q1$			$Q2$			$Q3$			$Q4$			$Q5$		
++ (5)	2	1/1	2	1	1	3				3					
+ (4)	3	1/2	2	2	2/3			1/2	1	1	2	2		1/1	
o (3)		1		1			4	2	2	1	2		2	1	1
− (2)				1			1	1			1/1	1			1
−− (1)													1		
n/a						1			1			1	2	1/2	2
∅	4,4	4,2	4,5	3,6	4,2	5,0	2,8	3,3	3,3	4,4	3,0	3,3	2,3	3,7	2,5

Table D.3.: Results of the questions about usability principles $Q1$–$Q5$.

	Usability Principles														
	$Q6$			$Q7$			$Q8$			$Q9$			$Q10$		
++ (5)	1		1		1		1		1			1			
+ (4)	2	2/2	1		1/2	2		3	3	1	2		1	2/1	1
o (3)	2	1		1		1	1	3		2			2	1/1	1
− (2)			1	4			1								
−− (1)							2								
n/a		1	1		1/1	1				2	1/3	3	2	1	2
∅	3,8	3,8	3,7	2,2	4,3	3,7	2,4	3,5	4,3	3,3	4,0	5,0	3,3	3,6	3,5

Table D.4.: Results of the questions about usability principles $Q6$–$Q10$.

	Usefulness for Education																	
	$Q1$			$Q2$			$Q3$			$Q4$			$Q5$			$Q6$		
++ (5)	1	1/1	3		1/1	1		1/3	2	1	1/2	1	1	1			1	2
+ (4)	1	2/2	1	2	2/1	2	3	2	2	2	1/1	3		2/2	2	2	1/3	1
o (3)	3			3	1		1			2	1		2			3	1	
− (2)																		
−− (1)																		
n/a						1	1						2	1	2			1
∅	3,6	4,3	4,8	3,4	4,2	4,3	3,8	4,7	4,5	3,8	4,3	4,3	3,7	4,2	4,0	3,4	4,0	4,7

Table D.5.: Results of the questions about usefulness for education $Q1$–$Q6$.

	Novel Concepts & Future Ideas														
	$Q1$			$Q2$			$Q3$			$Q4$			$Q5$		
++ (5)	2	1/1	1	2	2/3	3	3	3/1	3	5	1/3	1	2	1	2
+ (4)	2	2/1	3	2	1	1	1	2	1		2		2	2	1
o (3)				1			1					2	1	1	
− (2)														2	1
−− (1)												1			
n/a	1	1													
∅	4,5	4,4	4,3	4,2	4,8	4,8	4,4	4,7	4,8	5,0	4,7	3,0	3,8	3,3	4,0

Table D.6.: Results of the questions about novel concept & future ideas $Q1$–$Q5$.

	General Questions					
	$Q6$			$Q7$		
yes	4	3/2	3	3	1/1	3
no		1			1/2	1
n/a	1		1	2	1	

Table D.7.: Results of the questions about novel concept & future ideas $Q6$–$Q7$.

Selbstständigkeitserklärung

Ich erkläre, dass ich die eingereichte Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Rostock, 31. März 2009

Angela Brennecke

Résumé

Personal Data

Name	Angela Brennecke
Place of Living	Rostock
Date of Birth	January 23, 1978
Place of Birth	Braunschweig, Germany
Nationality	German

Education

since 10/2007	Ph. D. student at the Department of Computer Graphics, University of Rostock
01/2005–09/2007	Ph. D. student at the Department of Simulation and Graphics, Otto–von–Guericke University of Magdeburg
09/2004	Diplom-Ingenieur (Dipl.-Ing.) in Computervisualistik (Computational Visualistics)
10/2000–09/2004	Graduate studies at the Faculty of Computer Science, Otto–von–Guericke University of Magdeburg
10/1998–07/2000	Undergraduate studies at the Faculty of Computer Science, Otto–von–Guericke University of Magdeburg
06/1997	Abitur (high-school diploma)
08/1990–06/1997	Gymnasium Neue Oberschule Braunschweig

Thesis Statements

1. The research fields of educational software and computer games have long been pursued almost separately. The lack of a common research background has prevented uniting the requirements of education with the engaging potential of computer games. Novel approaches to the design of digital game-based training systems are required that balance education and game playing appropriately.
2. The research field of digital game-based learning pursues the overall idea of supporting education with digital game-based training systems like serious games. Serious games interactively involve the player in the game world and keep the player's motivation to continue playing at a high level throughout the game. This is a great advantage. Yet serious games only focus on learning situations and do not consider *education* as the actual field of application. As such they are nothing but learning enhancements that do not meet the overall requirements of education.
3. The introduction of digital game-based learning strategies to education can only be fruitful if digital game-based training systems support all elements of education. Education is conventionally comprised of six central elements: *teacher*, *student*, and *learning content* as well as the relationships of *teaching*, *learning*, and *communication & interaction*. These have to be technically represented by future digital game-based training systems.
4. A general framework for digital game-based training systems was developed that supports all these elements of education. The framework addresses teachers and students as game players and establishes learning content in a game story context. Both users are assigned roles as opponent or accomplice game player to reinforce the game playing experience as well as the educational experience. Teaching, learning, and communication & interaction are explicitly reflected in the framework's authoring, training, and reviewing components, which target specification, experience, and review training sessions. As a result, teachers and students have the possibility to engage with learning content in a game story context and can pursue traditional educational situations at the same time.

5. Teachers generally can be assumed to be unfamiliar with playing computer games and particularly unfamiliar with authoring training content. The proposed interaction paradigm of *in-game authoring* turns the authoring process into a game playing experience. By playing the game, the teacher can specify training sessions and gets easily acquainted with the training system.

6. Students often discredit the educational use of serious games because the applications seldom link back to the lessons and sometimes favor fiction over facts. The proposed interaction paradigm of *in-game training* targets authentic data presentation and the integration of external learning materials into digital game-based training systems to substantiate their educational use. The idea of playing against the teacher is considered motivating students to engage with the training systems.

7. The novel concept of a visual review log provides a comprehensible summary of training sessions in a short story or overview illustration. It serves as a basis for classroom discussion as well as for the teacher's evaluation or self-evaluation. The visual review log is a key element to facilitate the integration of digital game-based training systems into education because it directly supports teacher-student-communication.

8. The implementation of a digital game-based training system that is based on the general framework can be highly complex. A precise workflow schedule was formed to the specific requirements of digital game-based training systems that supplements the general framework. The workflow schedule provides a detailed guideline for developers that goes beyond abstract instructional design patterns.

9. The development and implementation of a digital game-based training system for crime scene investigation (CSI) training substantiated the applicability of the general framework and workflow schedule. Called *OpenCrimeScene*, the system reproduces typical situations of CSI education, which are situated in a "cop-&-robber" game story. The implementation of *OpenCrimeScene* resulted in several specific solutions to player character behavior in an interactive 3D virtual environment like implicit trace transfer and crime scene photography.

10. The general framework for digital game-based training systems can also be shifted to other software technologies. In particular, the concept of smart meeting rooms provides an adequate environment for applying the general framework. There, it could be used to control technological assistance in a practical training scenario.